TowerDebias: A Novel Debiasing Method Based on the Tower Property

Student: Aditya Mittal

University of California, Davis

Department of Statistics

adimittal@ucdavis.edu

Instructor: Norm Matloff

University of California, Davis

Department of Computer Science

nsmatloff@ucdavis.edu

Course: STA 194H - Special Studies for Honors Students

March 16, 2024

1 Abstract

In recent years, the rapid development of machine learning models across diverse sectors, such as business, healthcare, and legal systems, have become increasingly relevant. As these models play a pivotal role in critical decision-making processes with a broad range of consumer impact, a significant concern arises: the issue of social fairness in machine learning. The primary objective of fair machine learning is to mitigate biases linked to sensitive attributes that impact the predictive results of algorithms. In this paper, we introduce the towerDebias method, designed to eliminate the influence of sensitive variables on predictions produced by black-box machine learning models. This novel approach is guided by the Tower Property of conditional expectation, with its primary goal of improving the fairness of predictions during the post-processing stage with some potential loss in accuracy. Notably, this approach is widely flexible and does not require an intricate understanding of the underlying black-box algorithm itself. Empirical results from several fairML datasets demonstrate the efficacy of the towerDebias method in improving fairness at some expense of utility in both regression and classification settings. Our findings yields valuable insights into the fairness vs. utility tradeoff on the application of towerDebias across various contexts.

2 Introduction

Machine learning resides at the intersection of statistics and computer science, with its primary goal being to leverage data and algorithms to generate accurate predictions. This area holds significant prominence in the growing field of data science. Using statistical frameworks, a diverse range of algorithms have been developed to create predictive models intended for purposes such as classification and regression. [14] Moreover, these algorithms play a crucial role in deriving meaningful insights and ultimately contribute to informed decision making processes.

In recent years, the rapid development of machine learning models and their extensive commercial applications across many sectors including (but not limited to) business, healthcare, and legal systems have become increasingly prominent. [17] As these models begin to play an integral role in guiding critical decisions with potential repercussions for consumers on a large scale, a noteworthy concern emerges: the issue of fairness in machine learning. Fairness in algorithmic machine learning has become particularly noteworthy as it revolves around the correction of algorithmic biases linked to protected attributes across its predictive abilities. Protected attributes include variables that contain sensitive information regarding an individual: race, religion, gender, marital status, etc. [15] With this, the occurrence of disparate impact becomes apparent as certain protected classes are unintentionally favored or disfavored due to algorithmic biases in relation to those attributes. [12] To foster fairness in machine learning algorithms, it becomes necessary to examine and control the effects of underlying biases from these algorithms with respect to the protected variables.

The widespread deployment of commercial black-box machine learning algorithms is becoming increasingly common for public use, which adds another level of complexity as it necessitates careful consideration of both legal and ethical implications. A case in point is Northpointe's development of the COMPAS (Correctional Offender Management Profiling for Alternative Sanctions) algorithm, which was designed to assess a criminal's likelihood of recidivism. The algorithm faced scrutiny under *Pro-Publica's* analysis which revealed algorithmic bias against black defendants as opposed to their white counterparts. [2] Northepoint refutes *Pro-Publica's* assertion that their software program treated both black and white defendants equally. *Pro-Publica* maintains its position, substantiating their findings with statistical analysis. [1]

Notably, many of these commercial black-box models may incorporate the use of sensitive variables, prompting ethical and legal concerns. To address these potential legal and ethical concerns, it becomes imperative to explore strategies for the removal of the effects of these sensitive variables from black-box algorithms. In the pursuit of achieving fairness, there exists an inherent Fairness-Utility tradeoff: a balance between fairness and predictive accuracy. This tradeoff implies that as fairness is prioritized in an algorithm, accuracy tends to suffer. However, the extent of this tradeoff depends on the fairness metrics employed and the specifics of each individual case.

Contributions

In this paper, our proposed towerDebias method aims to eliminate the effect of sensitive variables from predictions created by black-box models in order increase fairness. Notably, this approach does not necessitate the user to have a thorough understanding underlying algorithms behind the black-box models. Rather, it eliminates the effect of sensitive variables S in the post-processing stage via the Tower Property of conditional expectation – to be elaborated further in the methodology section.

The towerDebias method is extremely versatile and can be used across various different applications. Consider a situation where a model is trained to predict Y from X, where X contains the sensitive variable S. For instance, in cases such as using a black-box model in judicial systems like COMPAS, integrating S into predictions may lead to legal and ethical challenges. With the towerDebias method, one can utilize predictions from the black-box models for new data while effectively mitigating the influence of S to produce fair predictions.

3 Literature Review & Methodologies

The initial portion of this section gives an overview of previous efforts in the field of fair machine learning and defines several accepted fairness criteria that have been used to demonstrate the effectiveness of the algorithms. The latter segment provides the mathematical underpinnings behind the towerDebias approach.

3.1 Existing Methodologies

To achieve fairness, much work has been proposed to address discrimination across different stages of the deployment process of machine learning algorithms. We may outline them as follows: [10]

1. **Pre-processing**: Pre-processing refers to the processing of data before training the algorithm to reduce the bias associated with sensitive features.
2. **In-processing**: In-processing is applied to the design and training of models to induce fairness.
3. **Post-processing**: Post-processing methods refer to the modification of the model's predictions after it has been trained.

The central idea surrounding individual fairness underscores the fact that *similar individuals should be treated similarly.* A notable approach to enhancing individual fairness can be shown via propensity score matching. This method allows for the pairing of similar and dissimilar based on causal analysis, allowing for evaluation of individual fairness and capture the notion of similar treatment in probabilistic classifiers. [8] In addition, one statistical approach to create fair machine learning models has been also been introduced via Marco Scutari's FairML package. [18] This approach involves penalizing regression results (ridge regression) to produce interpretable, fair models. To mitigate the effect of sensitive variables, the models incorporates an unfairness parameter, where 0 signifies perfect fairness and 1 indicates no fairness constraints. Scutari notes that attaining perfect fairness may not be feasible in practical settings, thus the users must find a balance between increasing fairness and maintaining accuracy. The package includes several functions: *Fair Ridge Regression Model*, *Zafar's Linear Regression* for regression and *Fair Generalized Ridge Regression model*, *Zafar's Logistic Regression* for binary classification. In particular, both fair ridge regression functions allow for additional flexibility by allowing for a non-convex optimization function with fairness constraints as proposed in Komiyama et al. [9] Notably, both of Komiyama and Zafar's models define fairness as statistical parity. To increase fairness, both fair regression functions aim to constrain the overall effect of S to make the model more fair. Alternatively, Zafar's approach aims to constrain the effects of the individual sensitive attributes S_i. [20] In this paper, we apply the towerDebias approach to the aforementioned functions from the FairML package to further mitigate the impact of S and increase fairness, potentially at some expense of accuracy. In our empirical study, we set the unfairness parameter to 0.1 (almost perfect fairness) as an initial step to establish high fairness, and subsequently examine our results by application of towerDebias.

3.2 Fairness vs Utility Criteria

The inherent tradeoff between fairness and utility is an extremely important concept in research surrounding fairness in machine learning. Already, much literature has been published discussing several measures of fairness, many of which are quite complex and require significant knowledge of the individual subject matter. The following fairness measures for have been commonly accepted:

1. **Statistical Parity**: This metric requires an equal likelihood for individuals in both marginalized and non-marginalized groups to be assigned to the positive class. It can be represented as:

$$Pr(\hat{Y} = 1|S = 0) = Pr(\hat{Y} = 1|S = 1) \tag{1}$$

In this equation, the probability of being assigned to the positive class (1) is identical for both Group $S = 0$ and Group $S = 1$, signaling equal fairness. [3]

2. **Equalized Odds**: This metric requires that the predictor $\hat{Y}$ will satisfy equalized odds with respect to the protected attribute S and outcome Y, if $\hat{Y}$ and S are independent conditional on Y.

$$Pr(\hat{Y} = 1|S = 0, Y = y) = Pr(\hat{Y} = 1|S = 1, Y = y), \quad y \in \{0, 1\} \tag{2}$$

For the outcome y = 1, this constraint requires that $\hat{Y}$ has equal true positive rates across the two demographics $S = 0$ and $S = 1$. For y = 0, the constraint equalizes false positive rates. [6]

4. **Correlation Coefficient** $\rho(\hat{Y}, S)$: In this paper, we use the following measure of fairness.

$$\rho(\hat{Y}, S) = \text{correlation between predicted Y and S} \tag{3}$$

To assess fairness between our predicted Y and S, we use Kendall's Tau correlation coefficient to measure the relationship between these variables. Kendall's Tau is a non-parametric measure of the strength and direction of association that exists between two variables measured on an ordinal scale. [16] The choice of using Kendall's Tau over Pearson correlation is used because of its flexibility. For instance, it holds an advantage by not necessitating the assumption of linearity and allows us to measure the relationship between two variables beyond just continuous cases when Y or S take categorical/binary forms. When dealing with categorical variable S, we use one-hot encoding to generate dummy variables for each level. Subsequently, we can compute the reduction in correlation for each specific level of S. For the purpose of measuring fairness, we want to get smaller values (closer to 0) to indicate that our predicted Y is not associated with the sensitive variables. In ranked columns, a negative relationship does not mean much so we use the absolute value of Kendall's Tau correlation to measure the reduction between predictions created by the black-box model and towerDebias. [5]

To assess utility, we utilize a holdout set to compute test accuracy from predictions generated by machine learning algorithms. In regression contexts, we use the mean absolute prediction error (MAPE), and for classification scenarios, we use misclassification rate.

3.3 Tower Property of Conditional Expectation

The *Tower Property* of conditional expectation is a key concept in probability theory often employed in the context of conditional expectations. This property can be articulated as follows: [19]

$$E(Y|X) = E[E(Y|X, S)|X] \tag{4}$$

The equation above signifies that the conditional expectation of Y given X is equivalent to the conditional expectation of Y given both S and X, conditioned solely on X. At a broader level, this formula represents the population-level expected value of Y given X. Sampling from the population allows for the estimation of these expectations. To illustrate, consider the following example: predicting LSAT scores Y, based on race S and a single predictor family income X. Suppose there is a new individual to be predicted with an income \$50,000, then the expression would be:

$$E(LSAT|Income) = E[E(LSAT|Income, Race)|Income] \tag{5}$$

To get our new prediction via this methodology, we first compute the average conditional mean of LSAT scores based on both race and income within the inner expectation. Then, we can condition our

LSAT scores again with income at $50,000 through the outer expectation. Consequently, our updated prediction for LSAT scores will be solely conditioned on income through, thereby eliminating the effect of S on our prediction by leveraging the Tower Property. In essence, this implies that our prediction will represent the average LSAT scores of individuals at the population level who share an income of $50,000.

3.4 Relation to towerDebias

The towerDebias method uses the Tower Property of conditional expectation to mitigate the effect of sensitive variables S on the predictions generated by a black-box model. Consider the example above again, and suppose we have acquired a dataset containing a random sample from the given population. In a sample, there might be few or no rows with exactly $50,000 income, so we take the average of rows of those with income NEAR $50,000.

In this case, the selection of k is the number of nearest neighboring rows to compute the average LSAT scores with. This choice of k is quite crucial to minimizing the extent of influence of the sensitive variable. For instance, a small k may lead to an overly narrow selection that might not cause significant reduction in the correlation, while a larger k may include rows that are too distant and not representative of the new data point being debiased. Therefore, the choice of k is quite important as it balances the need for a sufficient sample size with the necessity of proximity to the target datapoint.

3.5 Reduction in Correlation

Suppose we are predicting Y using a single feature X and a numeric S. For simulation purposes, let's assume (X,S,Y) follows a trivariate normal distribution. This implies that the joint distribution of (X,S,Y) is normal, and furthermore, the marginal distributions of each individual variables – X, S, and Y – are also normal. [7]

We are interested in finding the reduction in (Pearson) correlation between predicted Y and S when predicting Y from (X,S) (Case 1) versus just from X (Case 2). Note that the focus of this simulation is on assessing the reduction in Pearson correlation, as opposed to the Kendall's Tau correlation in towerDebias.

The choice of Pearson correlation in this case is motivated by the nature of the trivariate normal distribution. The Pearson correlation coefficient is suitable as it provides a measure of the strength of the linear relationship between two numeric variables. [4] It's inherent linearity properties in the numerator and denominator allow for additional simplification of our expression. Specifically, Pearson's numerator involves computing Cov(U, V) and exhibits bilinearity in random variables U and V. This property of bilinearity allows for the additional simplification of expressions by factoring out any constants and leveraging additivity properties. In contrast, Kendall's Tau lacks such linearity properties and makes it incapable of similar simplifications. With this, Pearson correlation is appropriate for this study as it allows us to derive and analyze a closed-form expression for the reduction of correlation between the two cases. This choice provides us with a clear and interpretable framework for examining the impact of predicting Y from both X and S versus predicting Y solely from X.

Note that the formula to compute Pearson correlation between predicted Y and S $\rho(\hat{Y}, S)$ is: [13]

$$\rho(\hat{Y}, S) = \frac{\text{cov}(\hat{Y}, S)}{\sigma(\hat{Y})\sigma(S)} \tag{6}$$

In first case, we may express the linear relationship to predict Y from X and S as:

$$Y_1 = \beta_0 + \beta_1 X + \beta_2 S + \epsilon_1 \tag{7}$$

Here, β_0 represents the intercept term, β_1 is the coefficient for X, β_2 is the coefficient for S, and $\epsilon_1 \sim \mathcal{N}(0, \sigma_1^2)$. Note that ϵ_1 is independent of the other predictor terms. [11]

For the second case, we may express the linear relationship to predict Y from X as:

$$Y_2 = \delta_0 + \delta_1 X + \epsilon_2 \tag{8}$$

Similarly, δ_0 represents the intercept term, δ_1 is the coefficient for X, and $\epsilon_2 \sim \mathcal{N}(0, \sigma_2^2)$. Again, ϵ_2 is also independent of the predictors. [11] Note that we have a different set of parameters in X for each case so Y_1 and Y_2 will consist of different predictions.

Furthermore, as we have a trivariate normal distribution, we have linearity of S with X as well. Our relationship between S and X may then be described as:

$$S = \alpha_0 + \alpha_1 X + \epsilon_3 \tag{9}$$

Here, α_0 represents the intercept term, α_1 is the coefficient for X, and $\epsilon_3 \sim \mathcal{N}(0, \sigma_3^2)$. ϵ_3 is also independent of the predictor X. [11]

We are interested in $\rho_{\text{reduc}}(\hat{Y}, S)$ between case I and case II, then our closed form expressed can be defined as:

$$\rho_{\text{reduc}}(\hat{Y}, S) = \rho(\hat{Y}_1, S) - \rho(\hat{Y}_2, S) \tag{10}$$

With all three models models established, we can now show the derivation of the closed form for $\rho_{\text{reduc}}(\hat{Y}, S)$.

Case 1: To compute $\rho(\hat{Y}_1, S)$, we are predicting Y from X and S:

$$Y_1 = \beta_0 + \beta_1 X + \beta_2 S + \epsilon_1 \tag{11}$$

We may define $\rho(\hat{Y}_1, S)$ as:

$$\rho(\hat{Y}_1, S) = \frac{\text{cov}(\hat{Y}_1, S)}{\sigma(\hat{Y}_1)\sigma(S)} \tag{12}$$

We will first compute $\text{cov}(\hat{Y}_1, S)$:

$$\begin{aligned}
\text{cov}(\hat{Y}_1, S) &= \text{cov}(\beta_0 + \beta_1 X + \beta_2 S, S) && (13)\\
&= \text{cov}(\beta_0, S) + \text{cov}(\beta_1 X, S) + \text{cov}(\beta_2 S, S) && (14)\\
&= 0 + \beta_1 \text{cov}(X, S) + \beta_2 \sigma^2(S) && (15)\\
&= \beta_1 \alpha_1 \sigma^2(X) + \beta_2 \sigma^2(S) && (16)
\end{aligned}$$

Since β_0 is a constant, the covariance $\text{cov}(\beta_0, S)$ is 0. Furthermore, the derivation of $\text{cov}(X, S)$ can be found below:

$$\begin{aligned} \text{cov}(X, S) &= \text{cov}(X, \alpha_0 + \alpha_1 X + \epsilon_3) && (17)\\ &= \text{cov}(X, \alpha_0) + \text{cov}(X, \alpha_1 X) + \text{cov}(X, \epsilon_3) && (18)\\ &= 0 + \alpha_1 \sigma^2(X) + 0 && (19)\\ &= \alpha_1 \sigma^2(X) && (20) \end{aligned}$$

Since α_0 is a constant, the covariance $\text{cov}(X, \alpha_0)$ is 0. Additionally, as previously stated, ϵ_3 is independent of X thus $\text{cov}(X, \epsilon_2)$ is also 0.

Then, $\rho(\hat{Y}_1, S)$ is equal to:

$$\rho(\hat{Y}_1, S) = \frac{\beta_1 \alpha_1 \sigma^2(X) + \beta_2 \sigma^2(S)}{\sigma(\hat{Y}_1)\sigma(S)} \tag{21}$$

Case 2: We are predicting Y from just X, our equation is:

$$Y_2 = \delta_0 + \delta_1 X + \epsilon_3 \tag{22}$$

We may define $\rho(\hat{Y}_2, S)$ as:

$$\rho(\hat{Y}_2, S) = \frac{\text{cov}(\hat{Y}_2, S)}{\sigma(\hat{Y}_2)\sigma(S)} \tag{23}$$

We will compute $\text{cov}(\hat{Y}_2, S)$:

$$\begin{aligned} \text{cov}(\hat{Y}_2, S) &= \text{cov}(\delta_0 + \delta_1 X, S) && (24)\\ &= \text{cov}(\delta_0, S) + \delta_1 \text{cov}(X, S) && (25)\\ &= 0 + \delta_1 \text{cov}(X, S) && (26)\\ &= \delta_1 \text{cov}(X, S) && (27)\\ &= \delta_1 \alpha_1 \sigma^2(X) && (28) \end{aligned}$$

Then, $\rho(\hat{Y}_2, S)$ can be written as:

$$\rho(\hat{Y}_2, S) = \frac{\delta_1 \alpha_1 \sigma^2(X)}{\sigma(\hat{Y}_2)\sigma(S)} \tag{29}$$

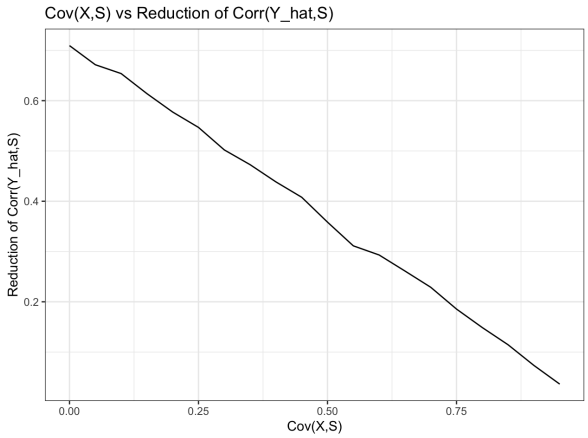
Then, $\rho_{\text{reduc}}(\hat{Y}, S)$ is:

$$\rho_{\text{reduc}}(\hat{Y}, S) = \rho(\hat{Y}_1, S) - \rho(\hat{Y}_2, S) \tag{30}$$

$$\rho_{\text{reduc}}(\hat{Y}, S) = \frac{\beta_1 \alpha_1 \sigma^2(X) + \beta_2 \sigma^2(S)}{\sigma(\hat{Y}_1)\sigma(S)} - \frac{\delta_1 \alpha_1 \sigma^2(X)}{\sigma(\hat{Y}_2)\sigma(S)} \tag{31}$$

The impact on overall $\rho_{\text{reduc}}(\hat{Y}, S)$ between the two cases is visually presented in the following graphs.

Effect of $Cov(X, S)$ on $\rho_{\textbf{reduc}}(\hat{Y}, S)$

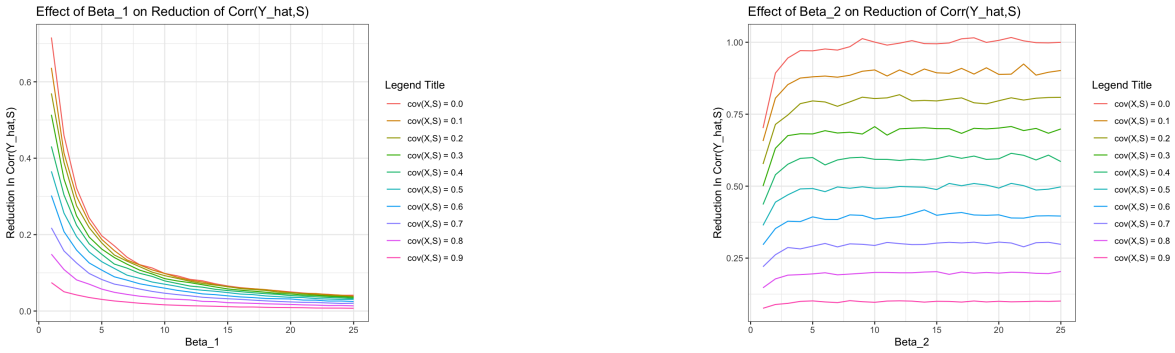


We simulated data for our predictor variables X and S using a multivariate normal distribution. Subsequently, we formed our response variable Y as a linear combination of X and S using various values for β_1 and β_2. Note that the marginal distribution of Y is also normal. Hence, the joint distribution of (X,S,Y) is trivariate normal. In terms of the marginal distributions of X and S, the value of μ for either variable did not exert any influence. However, variance and covariance between X and S had a significant impact on overall correlation reduction.

Figure 1: Cov(X,S) vs Reduction In Correlation

From the plot above, the relationship between overall reduction in correlation and varying values of covariance between X and S during the data generating process produces a negative linear trend. Notably, when the covariance between X as S is 0, the reduction of correlation between predicted Y and S across the two cases is quite significant (0.70 reduction). As the initial covariance between the X and S increases, holding other parameters fixed, the overall reduction in the correlation between predicted Y and S diminishes. This observation aligns with the expectation that X serves as a proxy for S when they are correlated. The graphs below detail the overall reduction in correlation between the two cases by trying different parameters through the data generating process.

Effect of β_i on $\rho_{\text{reduc}}(\hat{Y}, S)$

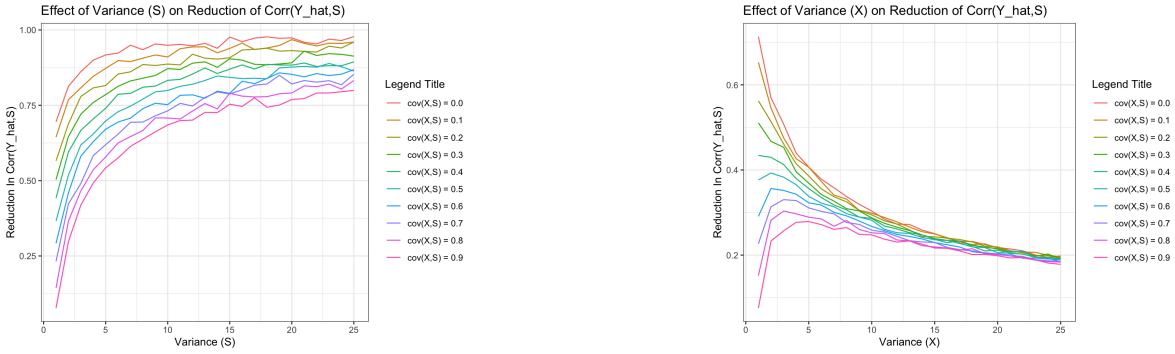


(a) β_1 vs. $\rho_{\text{reduc}}(\hat{Y}, S)$ (b) β_2 vs. $\rho_{\text{reduc}}(\hat{Y}, S)$

Figure 2: The results show effect of β_i across various values of $cov(X, S)$

As Y was generated as a linear combination of X and S via β_1 and β_2, the individual effects of these coefficients have an effect on $\rho_{\text{reduc}}(\hat{Y}, S)$. In the left plot, we hold β_2 fixed and create vary β_1 from 1:25 (also varying the initial covariance between X and S). Interestingly, when the initial cov(X,S) $= 0$ and $\beta_1 = 1$, the overall reduction in correlation between the two cases was the largest. As β_1 increases, the overall $\rho_{\text{reduc}}(\hat{Y}, S)$ between the two cases decreases. This trend continues across each initial value of $cov(X, S)$. In the right plot, holding β_1 fixed, we see the effect of increasing β_2 suggests higher reduction in overall correlation between the two. Thus, we can see the individual effect of each coefficient in generating Y and it's effect on the reduction of correlation between case I and II.

Effect of $\sigma^2(S)$ and $\sigma^2(X)$ on $\rho_{\text{reduc}}(\hat{Y}, S)$



(a) $\sigma^2(S)$ vs. $\rho_{\text{reduc}}(\hat{Y}, S)$ (b) $\sigma^2(X)$ vs. $\rho_{\text{reduc}}(\hat{Y}, S)$

Figure 3: The results show effects of $\sigma^2(X)$ and $\sigma^2(S)$ across various values of $cov(X, S)$

In addition to Y, the individual variance of both X and S have a signficant effect on overall correlation reduction. In the plot on the left, holding other parameters fixed, as variance S increases, we see that there is increased correlation reduction between predicted Y and S. Again this is true across each generated dataset with varying initial $cov(X, S)$. On the plot to the right, we see that increasing variance X indicates that the correlation reduction actually decreases between the two cases (as X becomes more spread). With this, the graphical results provide meaningful insights as to how different

parameters of the simulated trivariate normal data affects the reduction in correlation between case I and case II.

4 Empirical Study

In our empirical study, we implemented the towerDebias method across several widely recognized datasets in the domain of fair machine learning. Specifically, our empirical study leveraged five datasets: Svcensus, Law School Admissions, Compas, Iranian Churn, and Dutch Census.

Dataset	**Response Variable (Y)**	**Sensitive Variable(S)**	**Observations**	**Variables**
Svcensus	Wage Income (Continuous)	Gender (Binary: Male/Female)	20,090	6
Law Schools Admissions	LSAT Score (Continuous)	Race (Categorical: White, Black, Hispanic, Asian, Other)	20,800	11
Compas	Two-Year Recidivism (Binary: Yes/No)	Race (Categorical: White, Black, Hispanic)	5,497	15
Iranian Churn	Exited (Binary: 1/0)	Gender (binary: Male, Female), Age (Continuous)	10,000	11
Dutch Census	Occupation (Binary: 1/0)	Gender (binary: Male, Female)	60,420	12

Table 1: Datasets and Response/Sensitive Attribute Information

Across each dataset, we trained several different machine learning algorithms to produce baseline results in terms of both fairness and accuracy. The towerDebias function was then applied to each model's predictions in order to see the increase in fairness – i.e. reduction in $\rho(\hat{Y}, S)$ – at some potential cost of predictive performance. The study aims to understand how different choices of k nearest rows, selected during the application of the individual algorithm and dataset, contribute to the overall effectiveness of this approach.

In this empirical study, the towerDebias approach has been applied to various different conventional machine learning (ML) models, including Linear/Logistic Regression, K-Nearest Neighbors, XGBoost, and Neural Networks on both regression and classification settings. The *Quick And Easy Machine Learning* (qeML) package in R provided as a user-friendly framework for constructing models and generating predictions for Linear/Logistic Regression, K-Nearest Neighbors, and XGBoost. The Neural Network was trained using the Keras package in Python. The model featured three hidden layers with Relu activation functions, while the output layer employed either Relu or Sigmoid activation depending on context of the regression vs. classification task, respectively. Additionally, we also applied the towerDebias method to the fair models from the fairML package: Fair Ridge Regression (utilizing both algorithms from Scutari and Komiyama) and Zafar's Linear/Logistic Regression. Note that the fairML functions are designed to induce fairness (with unfairness parameter set at 0.1) within it's baseline predictions as compared to the conventional ML methods. The evaluation of the towerDebias approach involved examining the impact on fairness/utility versus k across each algorithm and dataset, and the results are presented through graphical representations.

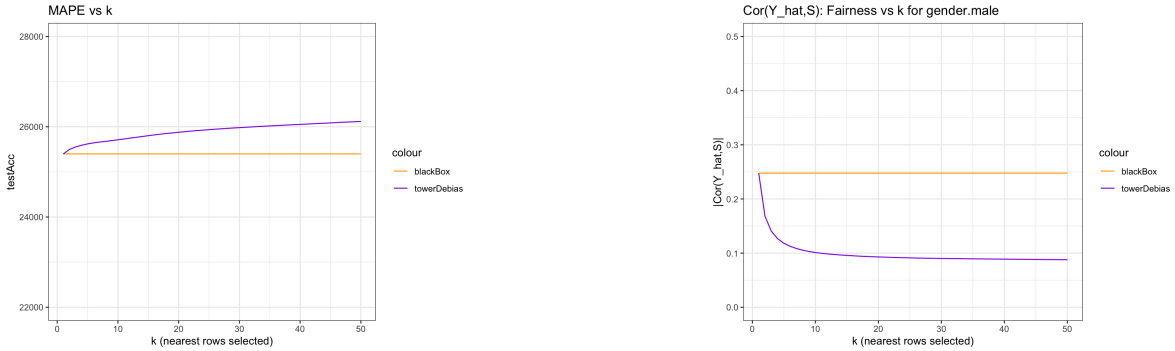
To reduce our results sampling variability from each experiment, we create 25 different holdout sets and compute the average test accuracy and correlation coefficient measures.

4.1 Svcensus

The svcensus dataset is a subset of US census data from back in early 2000, focusing on six different engineering occupations. Our response variable Y is predicting wage income, with respect to gender as the sensitive variable S. The graphs below displayed the application of the towerDebias method across the aforementioned algorithms.

Traditional ML Models vs. towerDebias

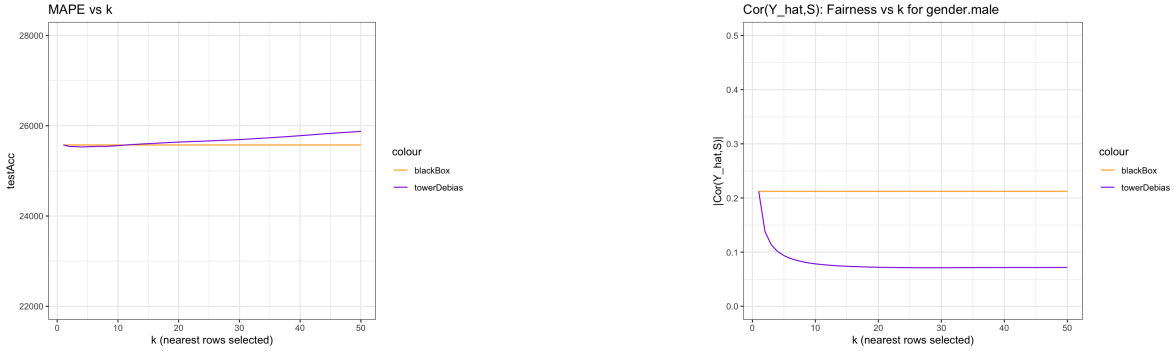
Linear Regression



(a) Mean Absolute Prediction Error vs k (b) Fairness vs k

Figure 4: Linear Regression vs TowerDebias: Fairness-Utility Tradeoff

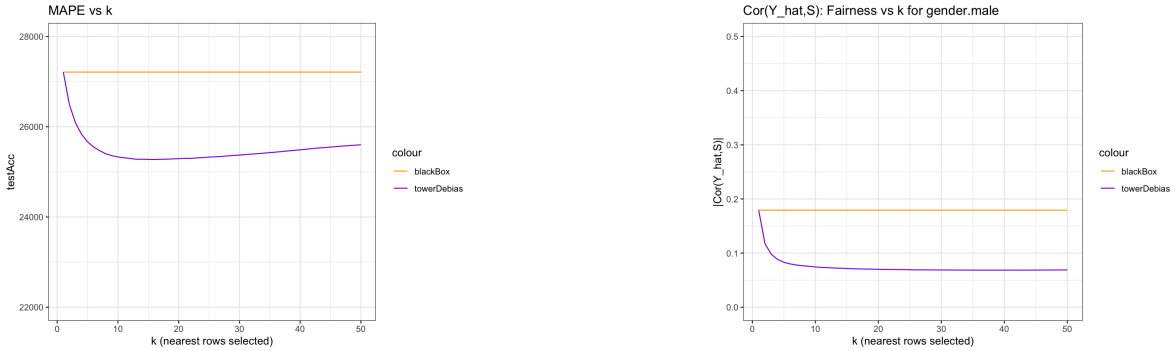
K-Nearest Neighbors



(a) Mean Absolute Prediction Error vs k (b) Fairness vs k

Figure 5: K-Nearest Neighbors vs TowerDebias: Fairness-Utility Tradeoff

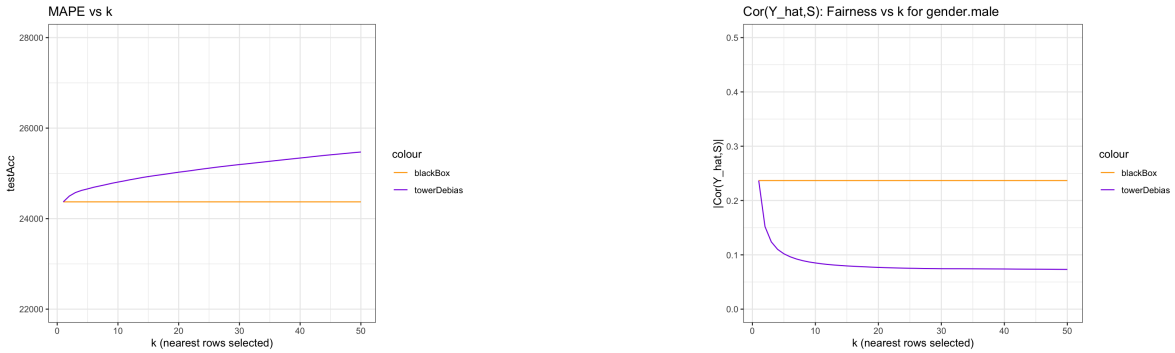
XGBoost



(a) Mean Absolute Prediction Error vs k (b) Fairness vs k

Figure 6: XGBoost vs TowerDebias: Fairness-Utility Tradeoff

Neural Network



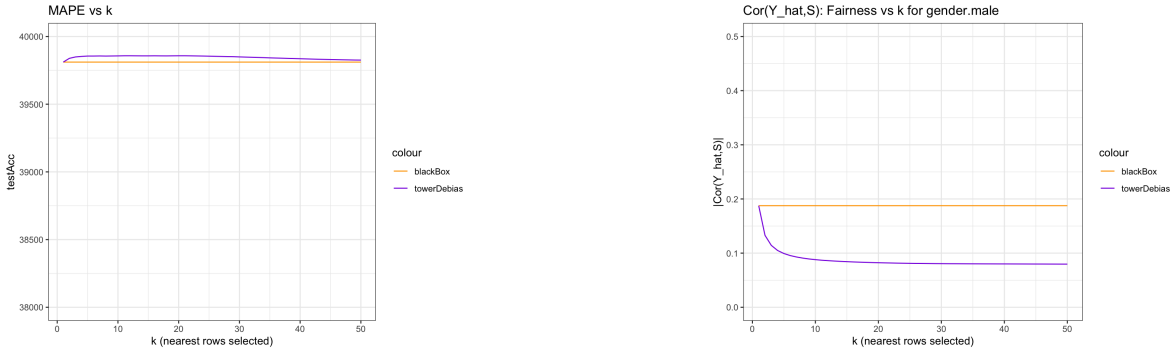
(a) Mean Absolute Prediction Error vs k (b) Fairness vs k

Figure 7: Neural Network vs TowerDebias: Fairness-Utility Tradeoff

The towerDebias method shows promising results in enhancing fairness against conventional ML models on the svcensus dataset. In linear regression, the baseline MAPE was approximately \$25,500 with a correlation $\rho(\hat{Y}, S)$ of 0.25. The application of towerDebias, as more k nearest rows were selected, displayed a controlled increase in MAPE while significantly decreasing the correlation, revealing its fairness-utility tradeoff. Notably, towerDebias significantly improved fairness as correlation dropped below 0.1 for k values exceeding 10. K-Nearest Neighbors (KNN) mirrored linear regression's trends, with a similar impact on fairness. The MAPE for KNN increased only after $k = 15$, indicating a smaller utility loss as compared to linear regression. The Neural Network also exhibited a parallel trend, showing a correlation reduction with increasing k rows. Interestingly, XGBoost had the highest initial MAPE at approximately \$27,300 and actually witnessed a decrease in both error rate and correlation when applying the towerDebias method – suggesting an improvement predictive power and fairness simultaneously.

Fair ML Models vs. towerDebias

Fair Ridge Regression (Scutari)

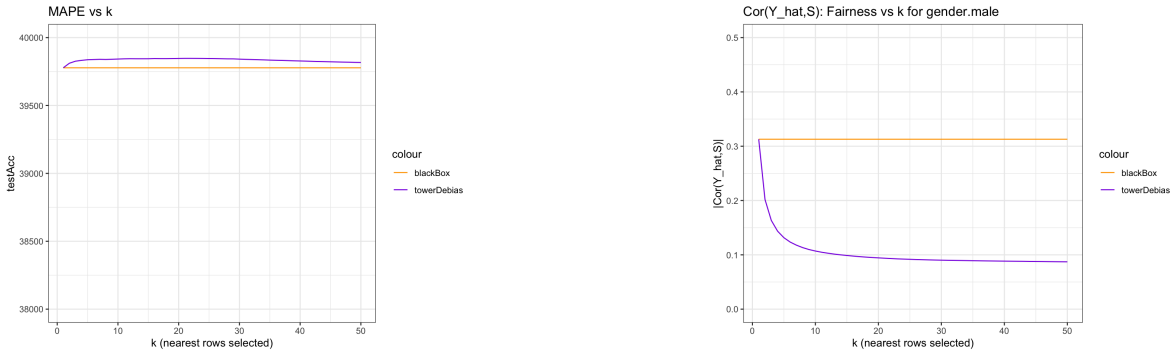


(a) Mean Absolute Prediction Error vs k

(b) Fairness vs k

Figure 8: FRRM (Scutari) vs TowerDebias: Fairness-Utility Tradeoff

Fair Ridge Regression (Komiyama)

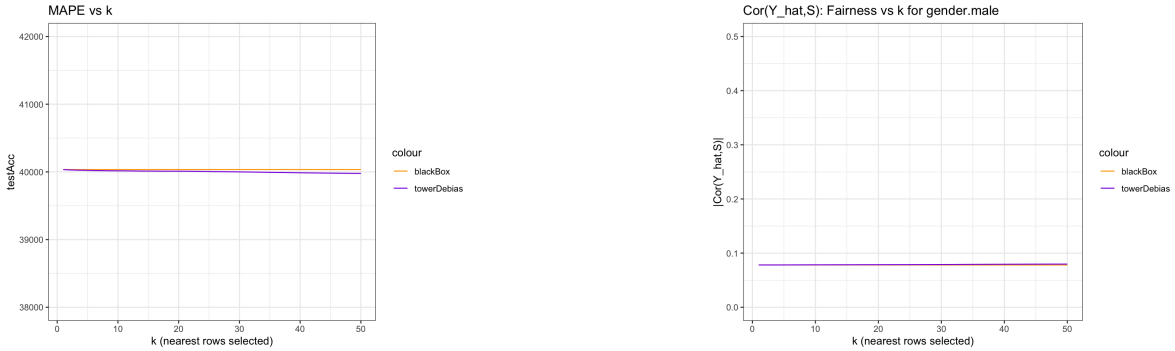


(a) Mean Absolute Prediction Error vs k

(b) Fairness vs k

Figure 9: FRRM (Komiyama) vs TowerDebias: Fairness-Utility Tradeoff

Zafar's Linear Regression



(a) Mean Absolute Prediction Error vs k

(b) Fairness vs k

Figure 10: ZLM vs TowerDebias: Fairness-Utility Tradeoff

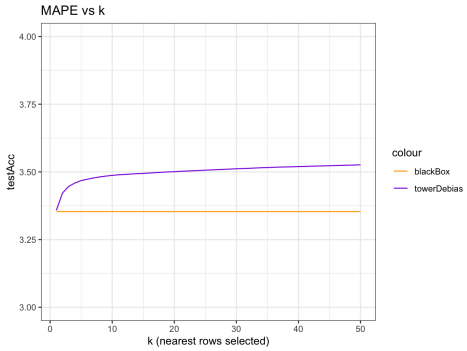
The application of towerDebias on fairML functions again reveals a consistent trend of increased fairness at a modest expense of utility. Notably, the initial Mean Absolute Percentage Error (MAPE) across all three models was notably higher at approximately $39,000. In both Fair Ridge Regression models, towerDebias effectively reduced the correlation between predicted income and gender, with a modest increase in MAPE. Zafar's linear regression model started with a lower initial correlation of 0.1, experienced a slight decrease in MAPE with towerDebias. Zafar's linear regression model displayed an initial correlation of 0.1, which was significantly lower than the other models. While the application of the towerDebias function did not result in a substantial reduction in correlation, there was a actually a slight decrease in the MAPE. Overall, applying the towerDebias method to svcensus data across various models demonstrates a significant decrease in $\rho(\hat{Y}, S)$, indicating improved fairness with a marginal impact on predictive accuracy. This underscores the towerDebias approach's advantage in addressing fairness concerns by mitigating the impact of sensitive variable S and creating fairer predictions.

4.2 Law Schools Admissions

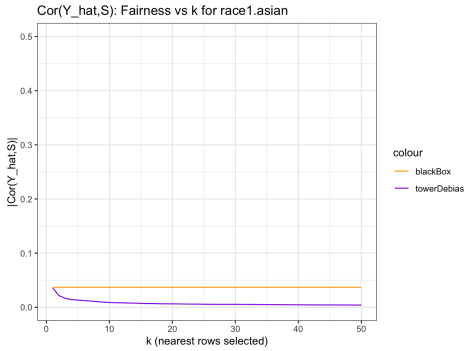
The law.school.admissions data is used to predict LSAT scores as the response variable Y with respect to race as sensitive variable S. The dataset included the following racial categories: Asian, Black, White, Other, and Hispanic. Note that the data concerns students who were admitted to law school, so in spite of the title, it's not about the admissions process itself.

Traditional ML Models vs. towerDebias

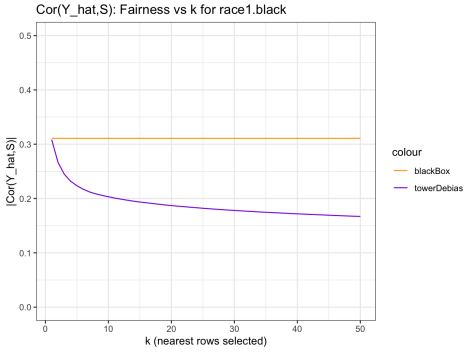
Linear Regression



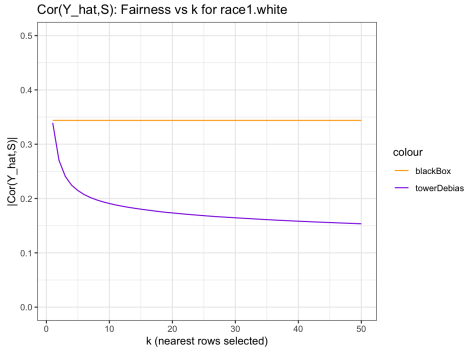
(a) Mean Absolute Prediction Error vs k



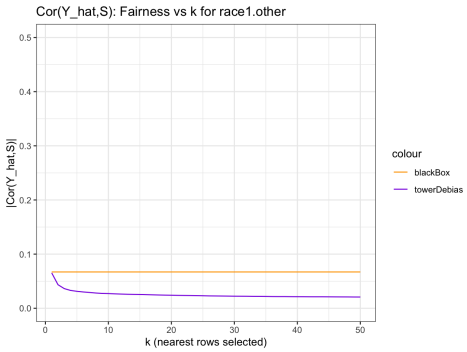
(b) Fairness vs k (Asian)



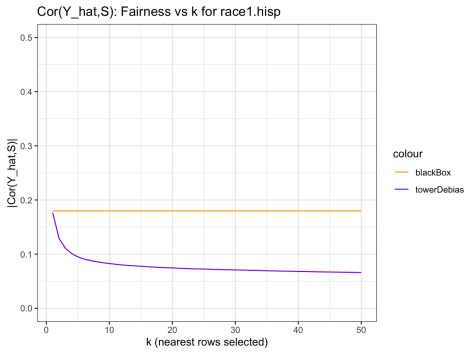
(c) Fairness vs k (Black)



(d) Fairness vs k (White)



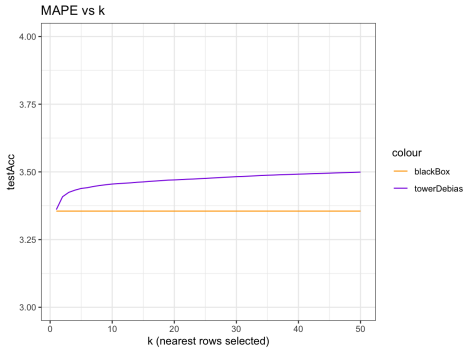
(e) Fairness vs k (Other)



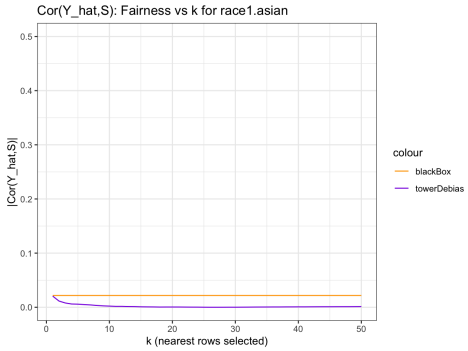
(f) Fairness vs k (Hispanic)

Figure 11: Linear Regression vs TowerDebias: Fairness-Utility Tradeoff

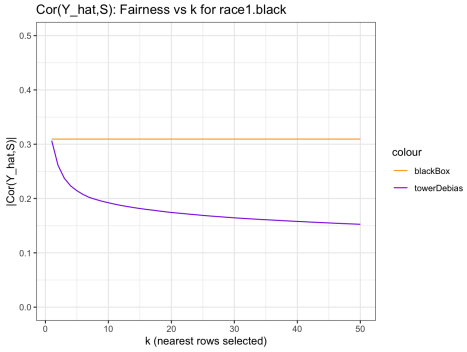
K-Nearest Neighbors



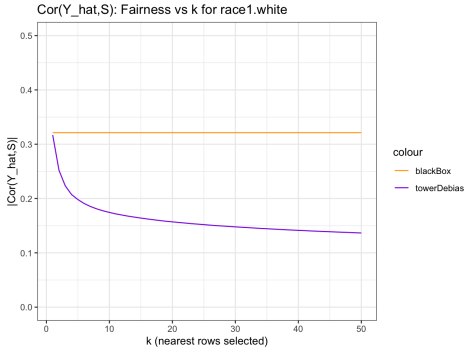
(a) Mean Absolute Prediction Error vs k



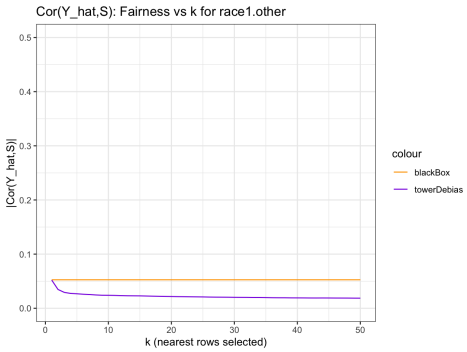
(b) Fairness vs k (Asian)



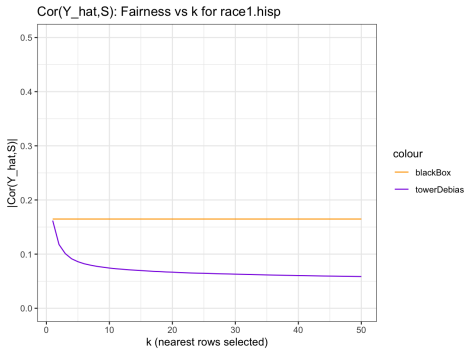
(c) Fairness vs k (Black)



(d) Fairness vs k (White)



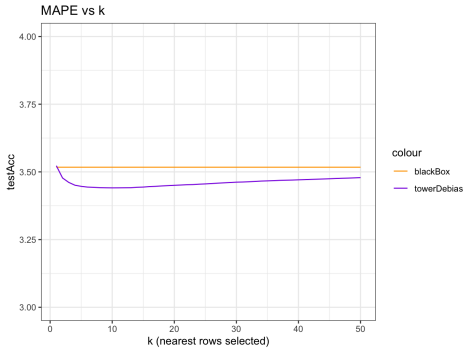
(e) Fairness vs k (Other)



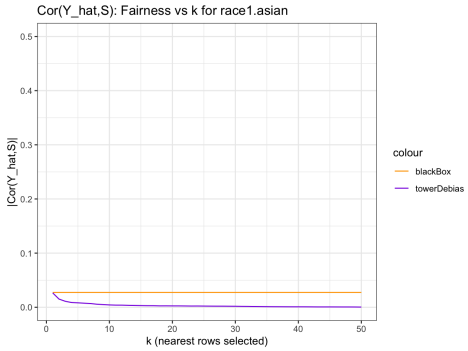
(f) Fairness vs k (Hispanic)

Figure 12: K-Nearest Neighbors vs TowerDebias: Fairness-Utility Tradeoff

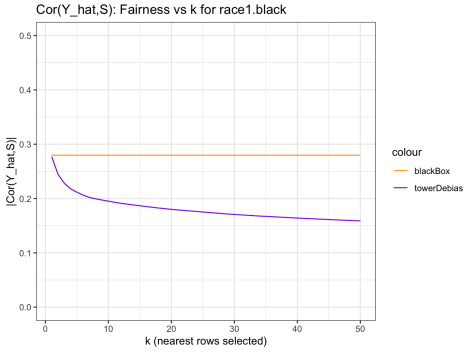
XGBoost



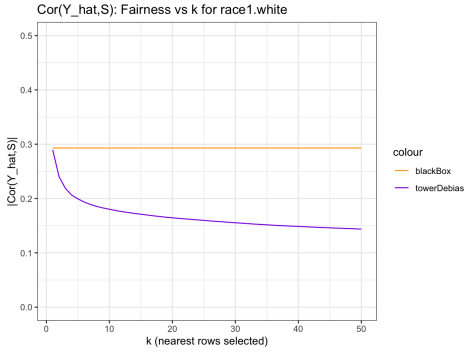
(a) Mean Absolute Prediction Error vs k



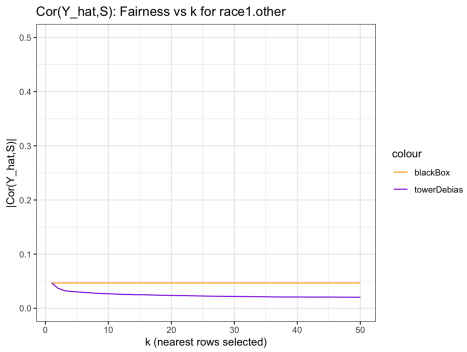
(b) Fairness vs k (Asian)



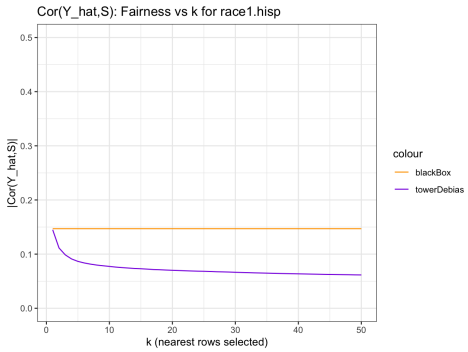
(c) Fairness vs k (Black)



(d) Fairness vs k (White)



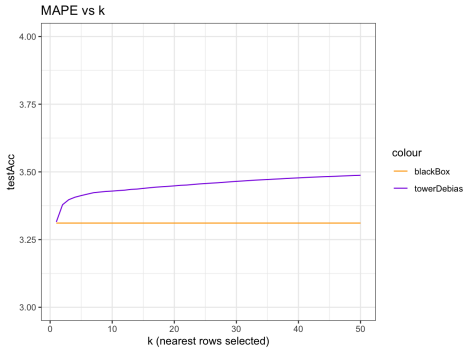
(e) Fairness vs k (Other)



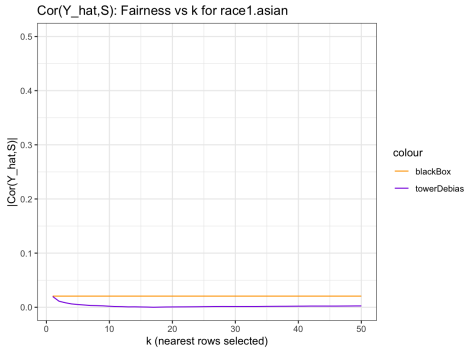
(f) Fairness vs k (Hispanic)

Figure 13: XGBoost vs TowerDebias: Fairness-Utility Tradeoff

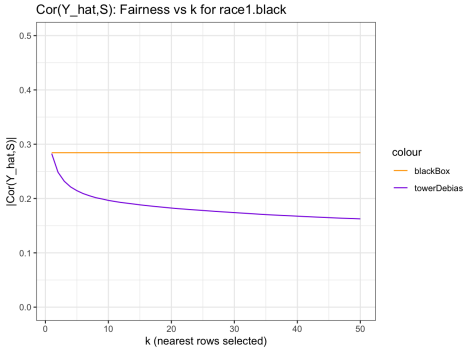
Neural Network



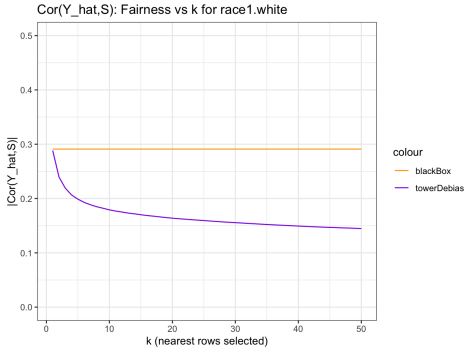
(a) Mean Absolute Prediction Error vs k



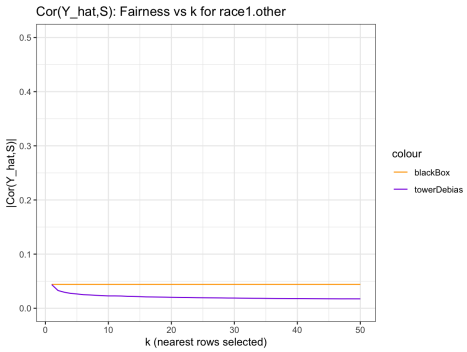
(b) Fairness vs k (Asian)



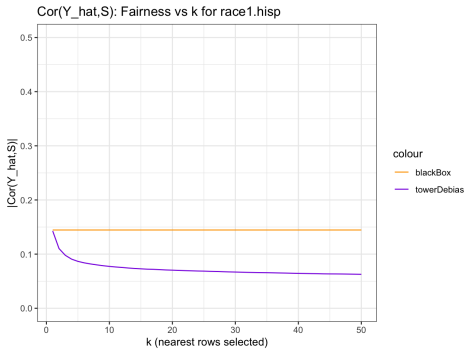
(c) Fairness vs k (Black)



(d) Fairness vs k (White)



(e) Fairness vs k (Other)



(f) Fairness vs k (Hispanic)

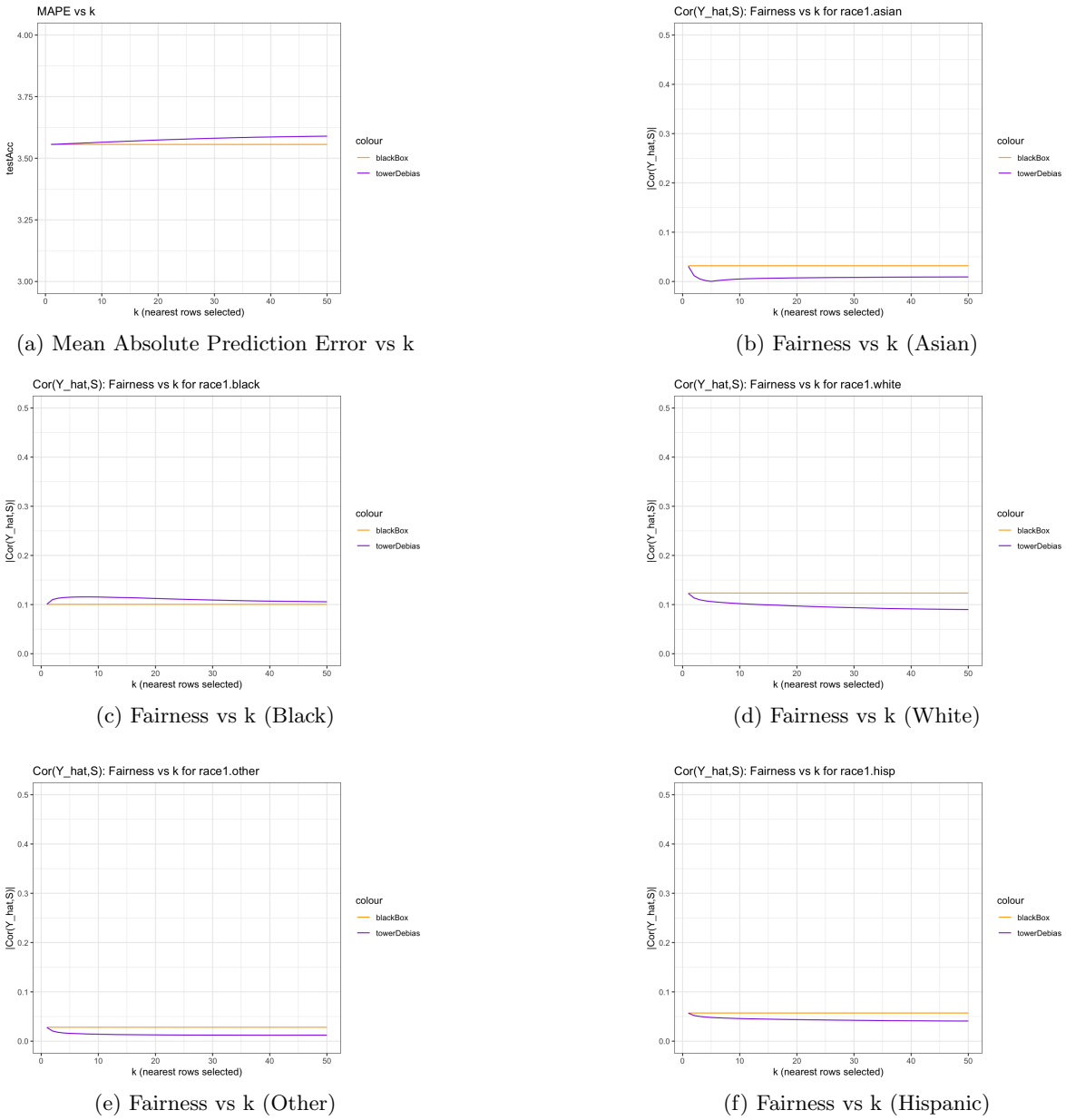
Figure 14: Neural Network vs TowerDebias: Fairness-Utility Tradeoff

In examining each method in the law schools data, the towerDebias approach consistently diminishes the impact of the sensitive variable across each racial group. In the context of linear regression, the initial Mean Absolute Percentage Error (MAPE) for LSAT scores is 3.35. As k rows increase, the error rate shows an increasing trend. Initial correlation $\rho(\hat{Y}, S)$ varies across races, but the application of the towerDebias method consistently reduces this coefficient. For example, as k increases, the correlation for African Americans drops from 0.3 to 0.18, and for whites, it decreases from 0.35 to 0.15 – indicating a substantial reduction. Both K-Nearest Neighbors (KNN) and Neural Network algorithms mirror the

results from linear regression, revealing increased fairness at the expense of some utility. Notably, the application of towerDebias to XGBoost produced similar results to svcensus data. The selection of k rows actually displayed slight decrease in both error rate and correlation, suggesting simultaneous enhancements in predictive power and fairness.

Fair ML Models vs. towerDebias

Fair Ridge Regression (Scutari)



(a) Mean Absolute Prediction Error vs k

(b) Fairness vs k (Asian)

(c) Fairness vs k (Black)

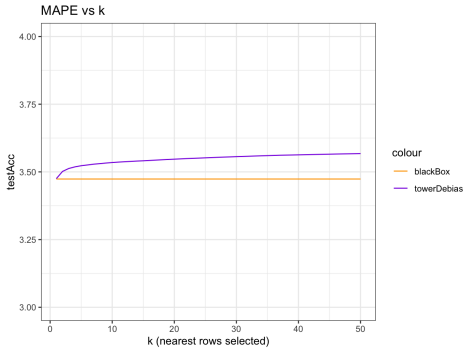
(d) Fairness vs k (White)

(e) Fairness vs k (Other)

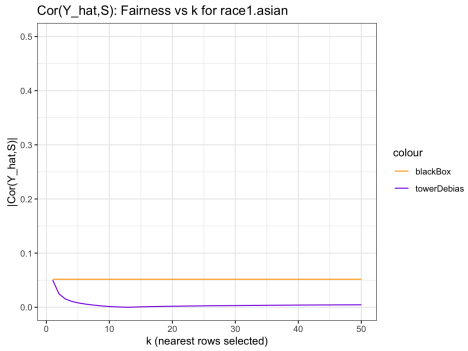
(f) Fairness vs k (Hispanic)

Figure 15: FRRM (Scutari) vs TowerDebias: Fairness-Utility Tradeoff

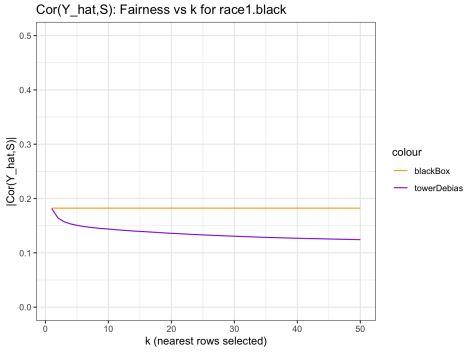
Fair Ridge Regression (Komiyama)



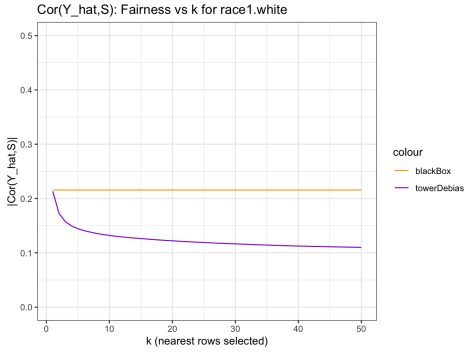
(a) Mean Absolute Prediction Error vs k



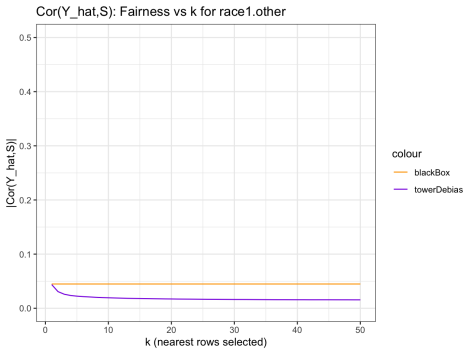
(b) Fairness vs k (Asian)



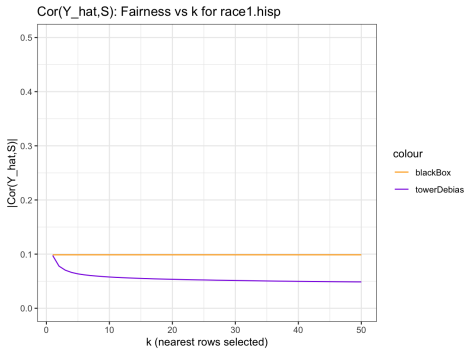
(c) Fairness vs k (Black)



(d) Fairness vs k (White)



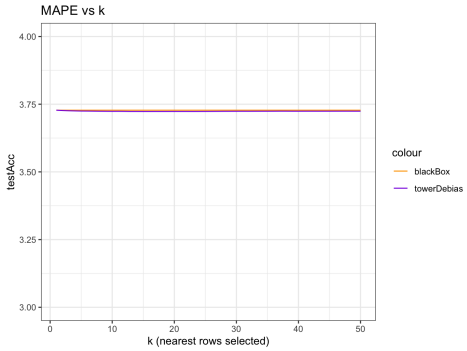
(e) Fairness vs k (Other)



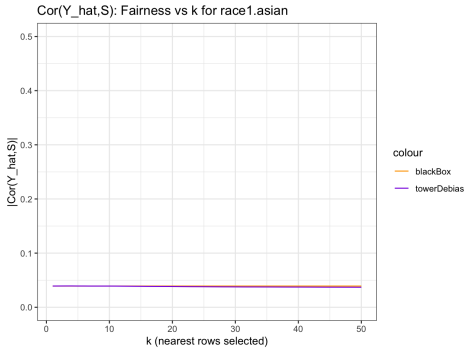
(f) Fairness vs k (Hispanic)

Figure 16: FRRM (Komiyama) vs TowerDebias: Fairness-Utility Tradeoff

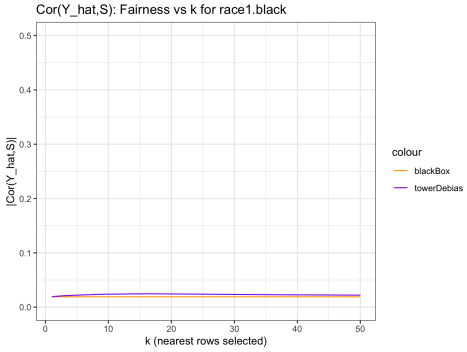
Zafar's Linear Regression



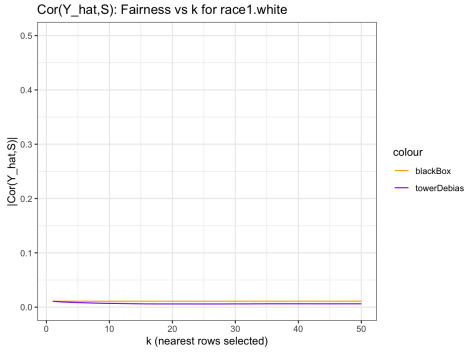
(a) Mean Absolute Prediction Error vs k



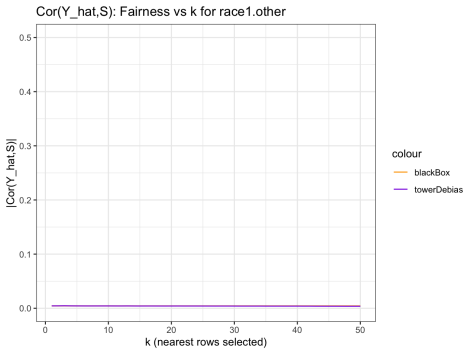
(b) Fairness vs k (Asian)



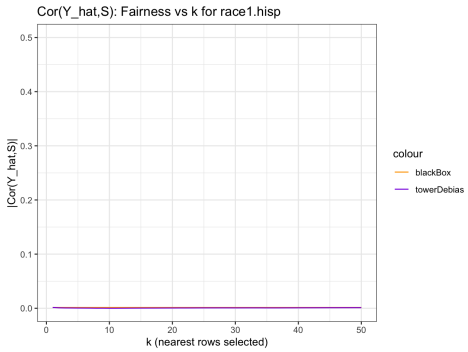
(c) Fairness vs k (Black)



(d) Fairness vs k (White)



(e) Fairness vs k (Other)



(f) Fairness vs k (Hispanic)

Figure 17: ZLM vs TowerDebias: Fairness-Utility Tradeoff

The application of the towerDebias method to fairML functions consistently enhances fairness across diverse racial groups in law school admissions data. Both Fair Ridge Regression models exhibit notably lower initial correlations across each racial category (as compared to the traditional ML models), with the towerDebias function further reducing correlations as the number of k nearest rows increase. Zafar's linear regression model maintains stability in MAPE without a substantial reduction in correlation with the application of the towerDebias function. Overall, the results highlight a significant decrease in $\rho(\hat{Y}, S)$ for each racial group, emphasizing the towerDebias approach's effectiveness in improving

fairness while maintaining a controlled impact on predictive accuracy in the context of law school admissions data.

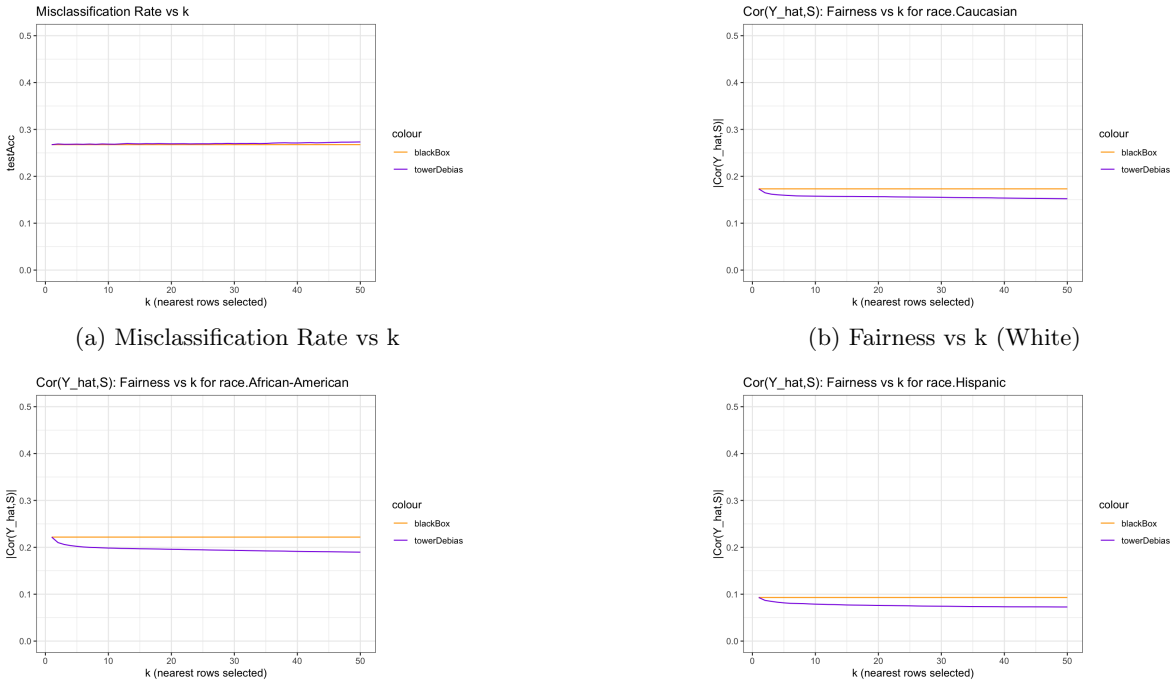
With the results from both datasets, the application of the towerDebias approach in regression settings appears highly promising. The consistent reduction in the impact of sensitive variables across various models and sensitive groups underscores the effectiveness of the towerDebias method in improving fairness at some loss of utility. The next step is to explore its application in classification cases.

4.3 Compas

The Compas dataset is used to predict whether a defendant is a recidivist Y, with race as the sensitive variable S. The dataset has been preprocessed to include three races: White, African American, and Hispanic. The graphs from applying the towerDebias method across various different models are displayed as following.

Traditional ML Models vs. towerDebias

Logistic Regression



(a) Misclassification Rate vs k

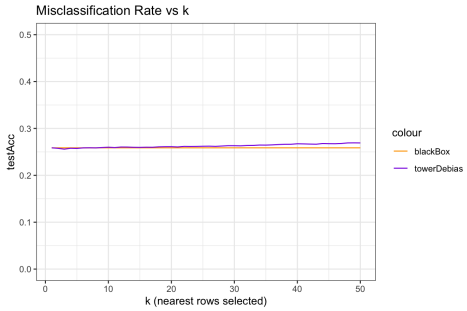
(b) Fairness vs k (White)

(c) Fairness vs k (Black)

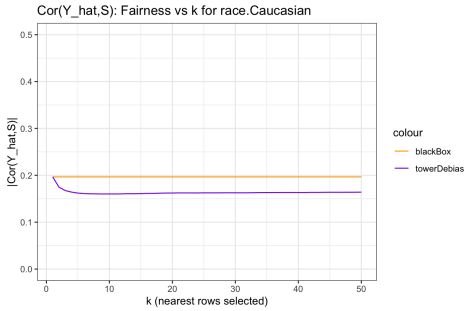
(d) Fairness vs k (Hispanic)

Figure 18: Logistic Regression vs TowerDebias: Fairness-Utility Tradeoff

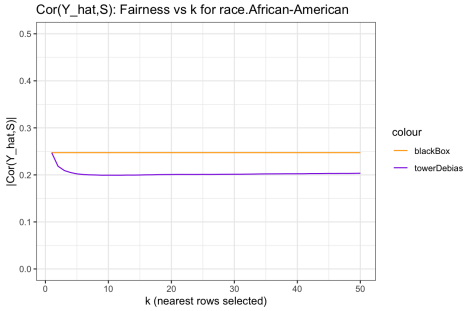
K-Nearest Neighbors



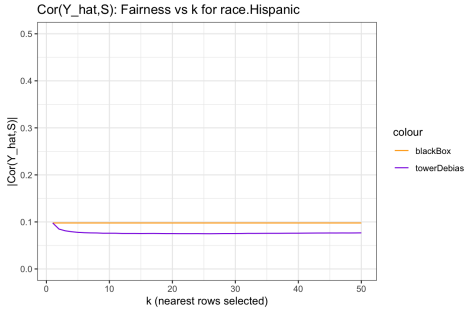
(a) Misclassification Rate vs k



(b) Fairness vs k (White)



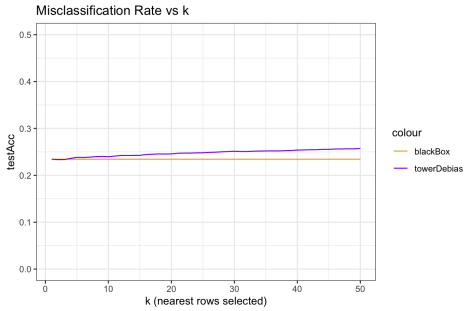
(c) Fairness vs k (Black)



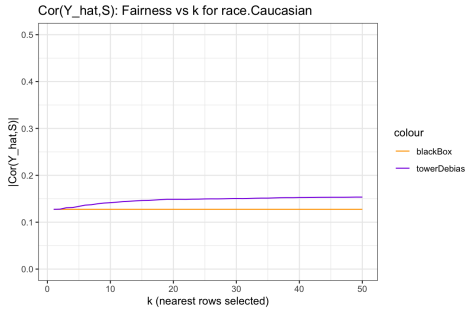
(d) Fairness vs k (Hispanic)

Figure 19: K-Nearest Neighbors vs TowerDebias: Fairness-Utility Tradeoff

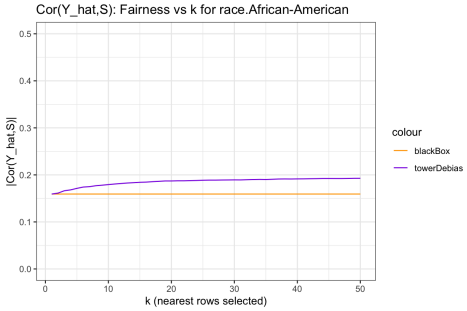
XGBoost



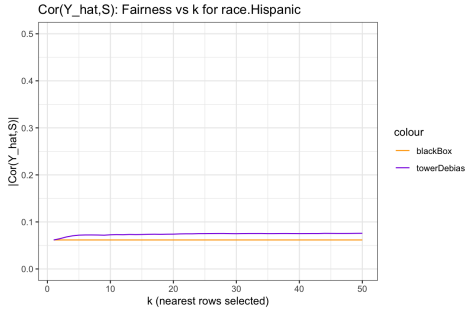
(a) Misclassification Rate vs k



(b) Fairness vs k (White)



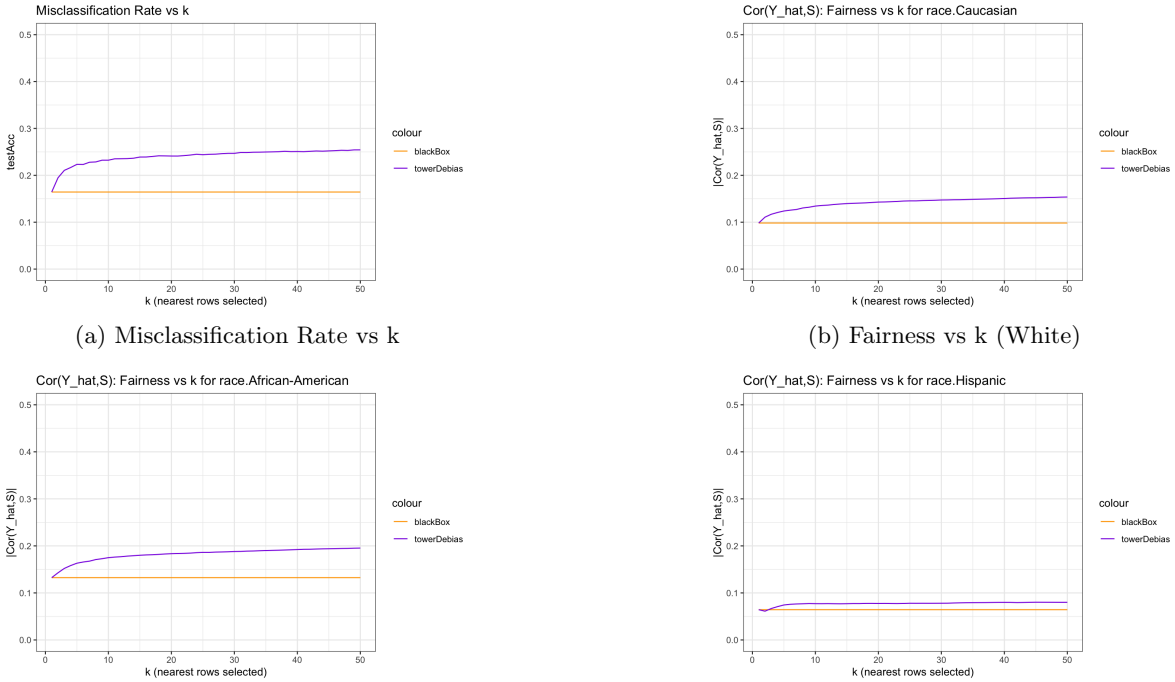
(c) Fairness vs k (Black)



(d) Fairness vs k (Hispanic)

Figure 20: XGBoost vs TowerDebias: Fairness-Utility Tradeoff

Neural Network



(a) Misclassification Rate vs k

(b) Fairness vs k (White)

(c) Fairness vs k (Black)

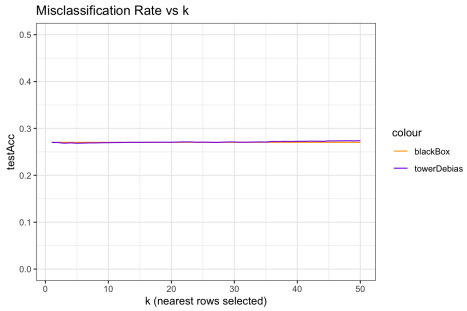
(d) Fairness vs k (Hispanic)

Figure 21: Neural Network vs TowerDebias: Fairness-Utility Tradeoff

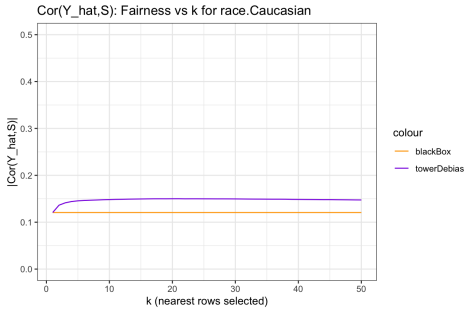
Analysis of the impact of towerDebias on individual algorithms within the Compas data reveals nuanced results. Logistic regression initially exhibited a misclassification rate of 0.27, with no increase in error rate upon applying towerDebias to an increasing number of k rows. The baseline correlation $\rho(\hat{Y}, S)$ varied across races, yet consistently reduced with the towerDebias method. For instance, as k increased, $\rho(\hat{Y}, S)$ for African Americans decreased from 0.23 to 0.2, with a comparable reduction for Caucasians and Hispanics. The KNN algorithm mirrored the results of logistic regression with application of towerDebias, demonstrating improved fairness at a tradeoff in utility. Interestingly, the towerDebias application on XGBoost and Neural Networks displayed unexpected fairness trends, showing an increase in $\rho(\hat{Y}, S)$ for each race concerning k. This deviation warrants further investigation to understand its underlying rationale.

Fair ML Models vs. towerDebias

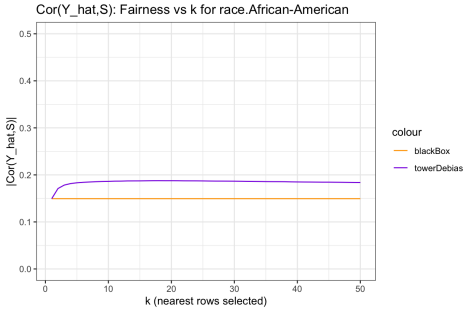
Fair Generalized Ridge Regression (Scutari)



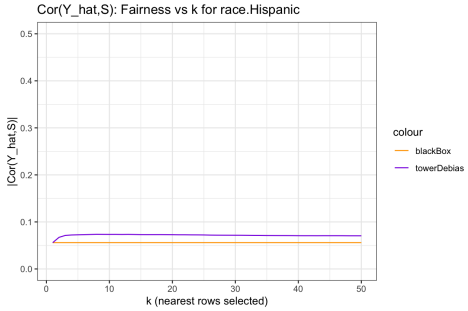
(a) Misclassification Rate vs k



(b) Fairness vs k (White)



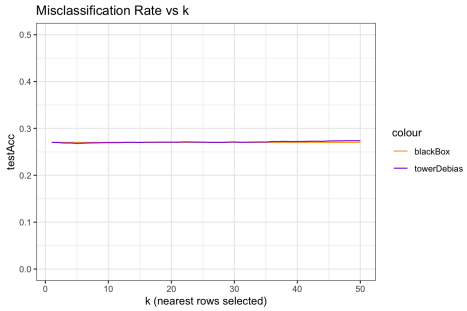
(c) Fairness vs k (Black)



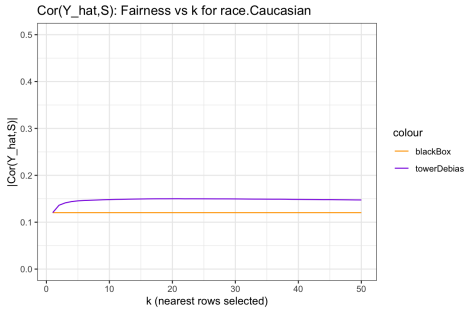
(d) Fairness vs k (Hispanic)

Figure 22: FGRRM (Scutari) vs TowerDebias: Fairness-Utility Tradeoff

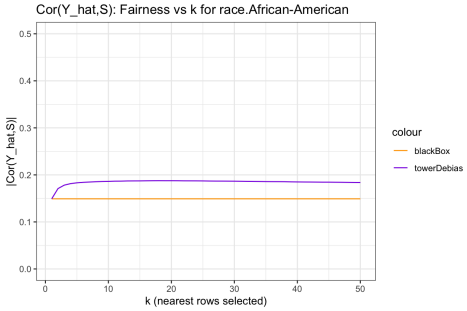
Fair Generalized Ridge Regression (Komiyama)



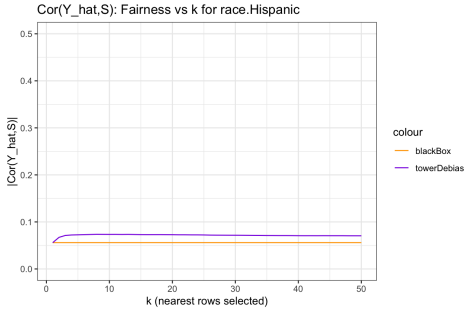
(a) Misclassification Rate vs k



(b) Fairness vs k (White)



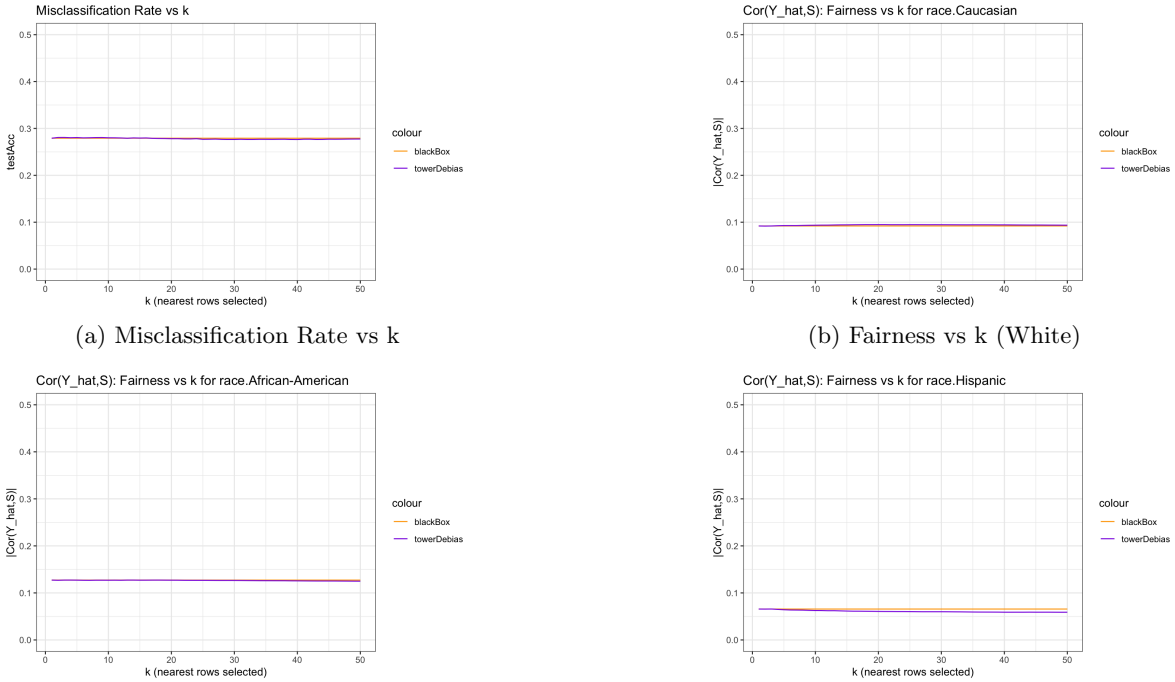
(c) Fairness vs k (Black)



(d) Fairness vs k (Hispanic)

Figure 23: FGRRM (Komiyama) vs TowerDebias: Fairness-Utility Tradeoff

Zafar's Logistic Regression



(a) Misclassification Rate vs k

(b) Fairness vs k (White)

(c) Fairness vs k (Black)

(d) Fairness vs k (Hispanic)

Figure 24: ZLRM vs TowerDebias: Fairness-Utility Tradeoff

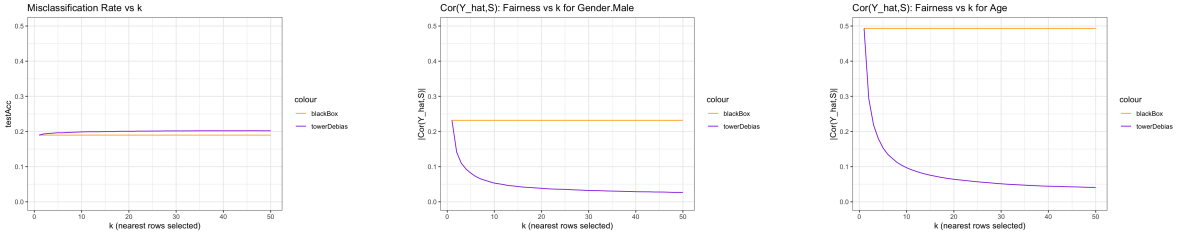
The fairness outcomes of towerDebias across fairML functions echo the unexpected patterns observed in the XGBoost and Neural Network algorithms. Contrary to expectations, $\rho(\hat{Y}, S)$ for each race increased concerning k. Both FGRRM model initially displayed smaller correlations than traditional ML models but showcased an upward trend with applying the towerDebias method. For example, the correlation for African Americans increased from 0.1 to 0.2, with Caucasians and Hispanics exhibiting a similar pattern. Zafar's Logistic Regression, while not resulting in a substantial reduction in correlation, maintained a stable misclassification rate without an increase. Despite the varied outcomes, the overarching theme indicates that the towerDebias method tends to trade-off a degree of utility for increased fairness in the context of the Compas data.

4.4 Iranian Churn

The Iranian Churn dataset is used to predict the discontinuation of a customer's relationship with a company using 'Exited' as the response variable Y with respect to 'Gender' and 'Age' as the sensitive variables S. The graphs from applying the towerDebias method across various different models are displayed as following.

Traditional ML Models vs. towerDebias

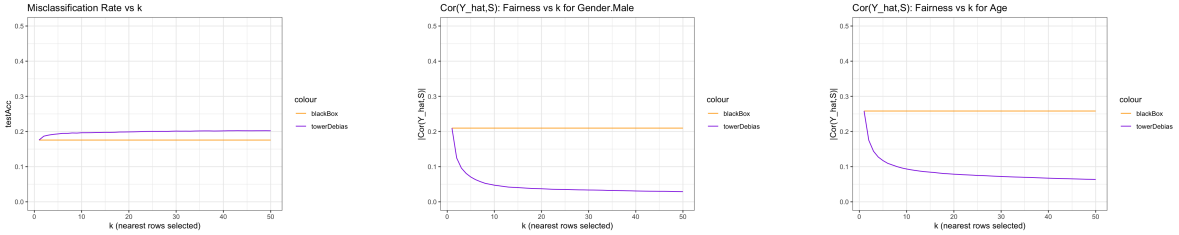
Logistic Regression



(a) Misclassification rate vs k (b) Fairness vs k (Gender) (c) Fairness vs k (Age)

Figure 25: Logistic Regression vs TowerDebias: Fairness-Utility Tradeoff

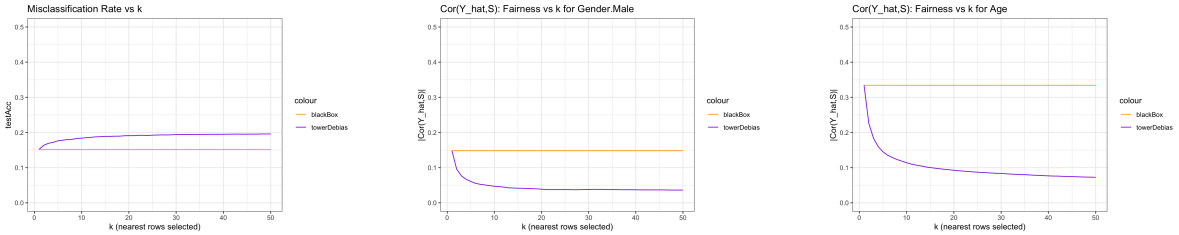
K-Nearest Neighbors



(a) Misclassification rate vs k (b) Fairness vs k (Gender) (c) Fairness vs k (Age)

Figure 26: K-Nearest Neighbors vs TowerDebias: Fairness-Utility Tradeoff

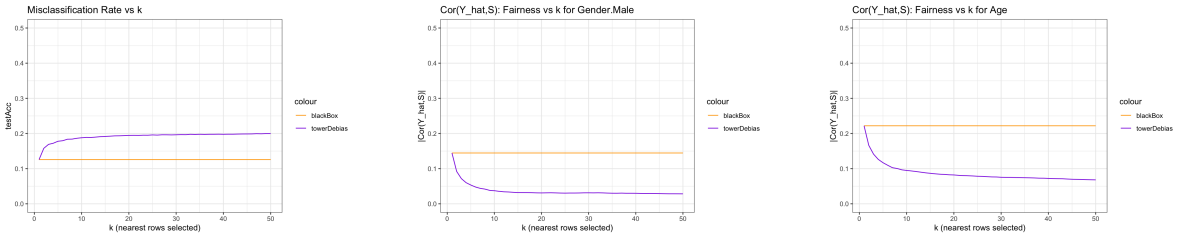
XGBoost



(a) Misclassification rate vs k (b) Fairness vs k (Gender) (c) Fairness vs k (Age)

Figure 27: XGBoost vs TowerDebias: Fairness-Utility Tradeoff

Neural Network



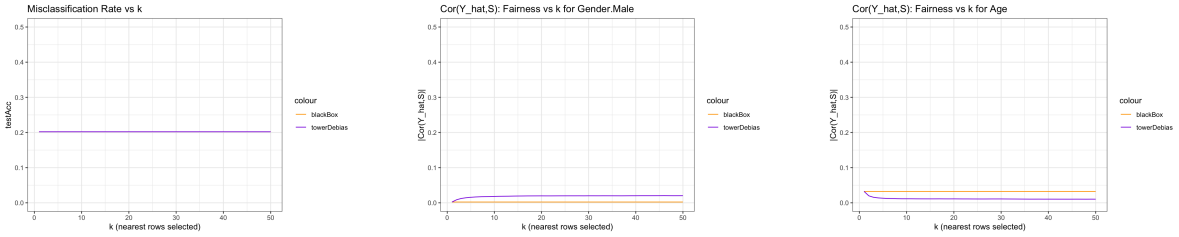
(a) Misclassification rate vs k (b) Fairness vs k (Gender) (c) Fairness vs k (Age)

Figure 28: Neural Network vs TowerDebias: Fairness-Utility Tradeoff

In our evaluation of the Iranian Churn dataset, the application of towerDebias across different machine learning algorithms yields promising results for classification. In logistic regression, the initial misclassificaiton rate stood just below 0.2 with $\rho(\hat{Y}, S)$ concerning gender and age at 0.25 and 0.5, respectively. Applying the towerDebias method, the error rate experienced a marginal increase to 0.2. The reduction in correlation is very significant for both sensitive variables as $\rho(\hat{Y}, S)$ decreased to 0.05, indicating a significant decline. The other methods KNN, XGBoost, and Neural Networks also displayed analogous behavior concerning the tradeoff between error and fairness. The application of towerDebias to all the algorithms demonstrated a notable decrease in $\rho(\hat{Y}, S)$ for gender and age. For instance, the KNN algorithm exhibits a gender correlation of 0.2, while the XGBoost and Neural Network algorithms shows a slightly lower correlation of 0.15. The towerDebias method produces a reduction of the correlation to below 0.05 (for gender), indicating a notable improvement in fairness. Additionally, this reduction in correlation was particularly pronounced for age, given the initial higher correlations. Ultimately, this emphasizes that towerDebias significantly enhances fairness with controlled effect on utility.

Fair ML Models vs. towerDebias

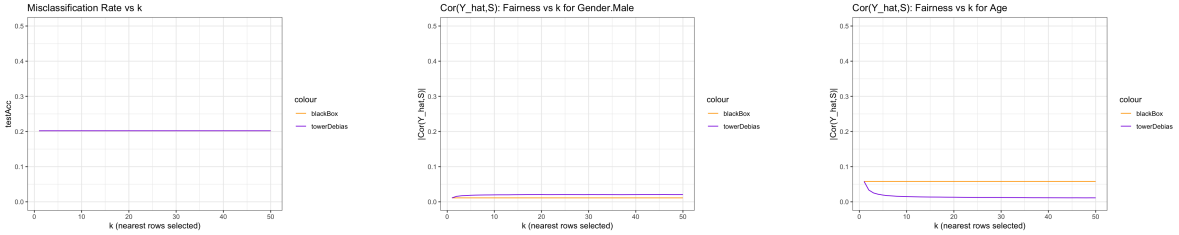
Fair Generalized Ridge Regression (Scutari)



(a) Misclassification rate vs k (b) Fairness vs k (Gender) (c) Fairness vs k (Age)

Figure 29: FGRRM (Scutari) vs TowerDebias: Fairness-Utility Tradeoff

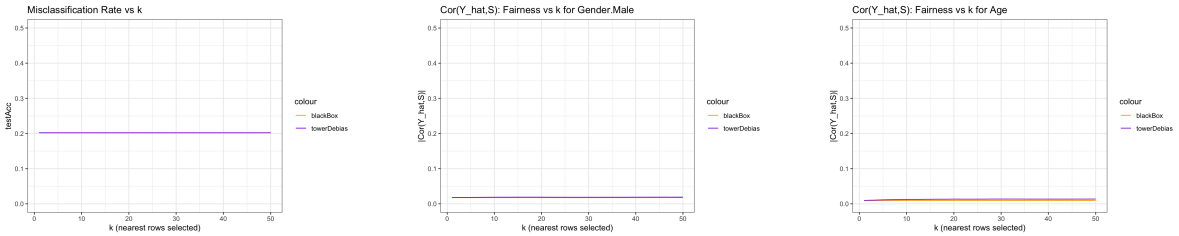
Fair Generalized Ridge Regression (Komiyama)



(a) Misclassification rate vs k (b) Fairness vs k (Gender) (c) Fairness vs k (Age)

Figure 30: FGRRM (Komiyama) vs TowerDebias: Fairness-Utility Tradeoff

Zafar's Logistic Regression



(a) Misclassification rate vs k (b) Fairness vs k (Gender) (c) Fairness vs k (Age)

Figure 31: ZLRM vs TowerDebias: Fairness-Utility Tradeoff

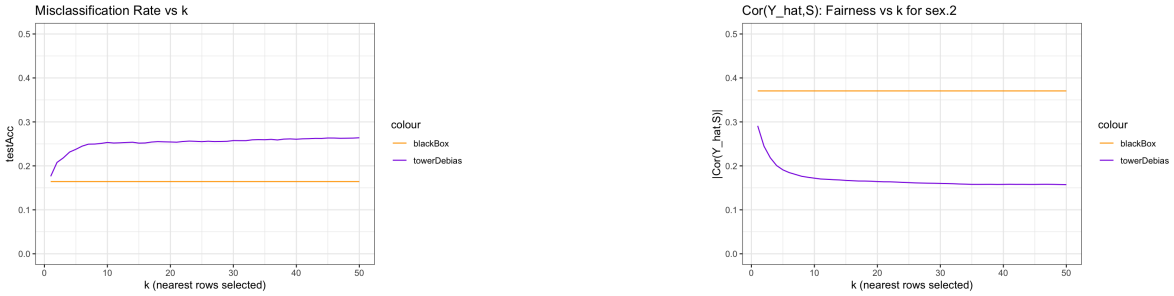
Both Fair Generalized Ridge Regression models displayed comparable error rate to traditional ML models, with notably smaller initial correlation coefficients for gender and age variables. As k increased, the towerDebias method demonstrated a reduction in correlation with age and a minimal increase with gender, showcasing its effectiveness in achieving fairness across the following sensitive variables. The towerDebias maintained the same misclassification rate as the FGRR model on this dataset, indicating no significant accuracy trade-offs. Applying towerDebias to Zafar's Logistic Regression did not lead to a substantial reduction in correlation, and it maintained a stable misclassification rate without an increase either. With this, the overall results indicate the effectiveness of the towerDebias method in maintaining the trade-off between degree of utility versus fairness in the context of the Iranian Churn data.

4.5 Dutch Census

The Dutch Census dataset was collected by the Dutch Central Bureau for Statistics in 2001. Our response variable Y is it predict whether or not someone has a prestigious occupation, with gender as the protected attribute S. The graphs from applying the towerDebias method across various different models are displayed as following.

Traditional ML Models vs. towerDebias

Logistic Regression



(a) Misclassification Rate vs k (b) Fairness vs K

Figure 32: Logistic Regression vs TowerDebias: Fairness-Utility Tradeoff

K-Nearest Neighbors

(a) Misclassification Rate vs k (b) Fairness vs k

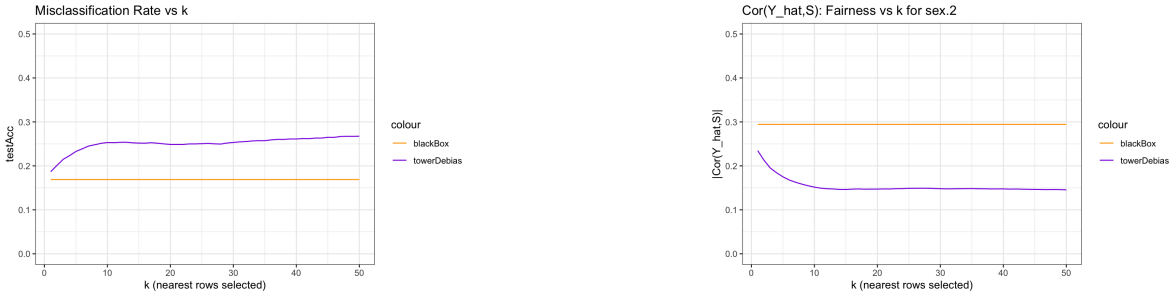
Figure 33: K-Nearest Neighbors vs TowerDebias: Fairness-Utility Tradeoff

XGBoost

(a) Misclassification Rate vs k (b) Fairness vs k

Figure 34: XGBoost vs TowerDebias: Fairness-Utility Tradeoff

Neural Network



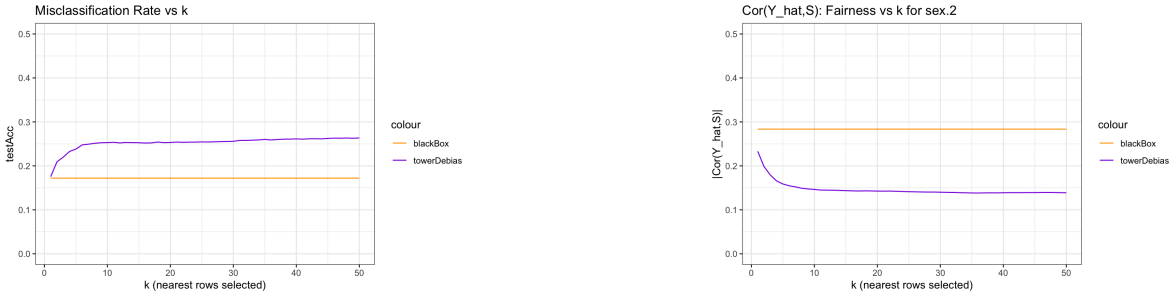
(a) Misclassification Rate vs k (b) Fairness vs k

Figure 35: Neural Network vs TowerDebias: Fairness-Utility Tradeoff

Empirical results from the Dutch Census dataset again underscore the effectiveness of towerDebias in classification tasks. In logistic regression, with a misclassification rate of approximately 0.18 and a correlation $\rho(\hat{Y}, S)$ of 0.35, applying towerDebias led to a marginal increase in the misclassification rate. However, the correlation consistently decreased as more k nearest rows were selected, emphasizing the inherent trade-off between fairness and utility. This consistent trend extended to other models, including K-Nearest Neighbors, XGBoost, and Neural Networks. Across all these methods, the application of the towerDebias method proves effective in enhancing the fairness of classification tasks compared to conventional ML models.

Fair Machine Learning Models vs. towerDebias

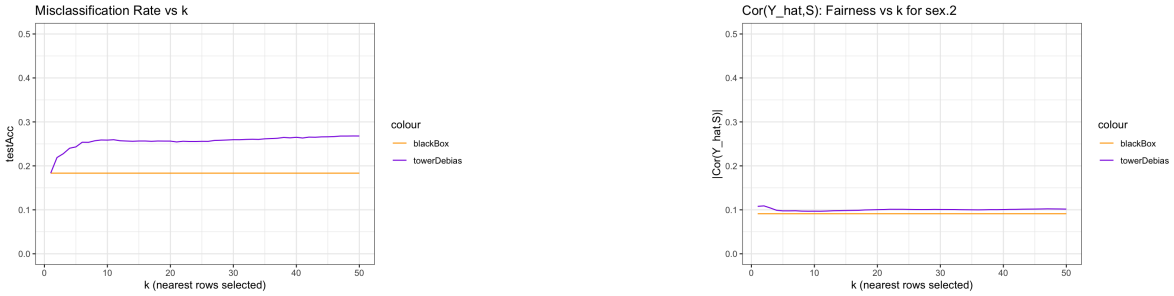
Fair Ridge Regression (Scutari)



(a) Misclassification Rate vs k (b) Fairness vs k

Figure 36: FRRM (Scutari) vs TowerDebias: Fairness-Utility Tradeoff

Fair Ridge Regression (Komiyama)



(a) Misclassification Rate vs k

(b) Fairness vs k

Figure 37: FRRM (Komiyama) vs TowerDebias: Fairness-Utility Tradeoff

Interestingly, Zafar's Logistic Regression incurred numerous errors and failed to produce predictions; hence, it has been excluded from our analysis. Examining both Fair Generalized Ridge Regression Models, Scutari's function, with the application of the towerDebias method, consistently reduces correlation as more k rows are selected. Conversely, the Komiyama function did not exhibit similar patterns, as the correlation appeared remain consistent despite the towerDebias method. Notably, for both FGRRM models, the error rate increased in relation to k, emphasizing the method's efficacy in balancing fairness versus utility once again.

Based on the insights from Compas, Iranian Churn, and the Dutch Census datasets, the application of the towerDebias method is quite effective on classification settings. The consistent decrease in the correlation of sensitive variables across different models emphasizes the efficacy of the towerDebias method in improving fairness, albeit at some cost of utility. Consequently, the towerDebias method presents promising results for both regression and classifications scenarios.

5 Discussions

The empirical examinations of the towerDebias method applied to a set of conventional machine learning algorithms yields promising outcomes. By employing this method, a notable reduction in $\rho(\hat{Y}, S)$ is achievable at a relatively modest loss in utility. Across multiple datasets and methodologies, the observed outcomes varied, but in the majority of cases, a discernible decrease in the correlation coefficient was evident. The empirical study emphasizes the effectiveness of different k values on the towerDebias approach. This choice significantly contributes to reducing the impact of sensitive variables and the effect on the utility. The results indicate the optimal choice of k varies depending on the specific data and algorithm being utilized.

In conclusion, the empirical evidence supports the efficacy of the towerDebias method as a valuable tool for promoting fairness in machine learning. The findings of this paper contribute to the ongoing research on fair machine learning and provide a foundation for further study and refinement of methods aimed at addressing algorithmic biases.

References

[1] Julia Angwin and Jeff Larson. Machine bias: Technical response to northpointe, 2016.

[2] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. Machine bias, 2016.

[3] Alycia Carey and Xintao Wu. The statistical fairness field guide: perspectives from social and formal sciences. *AI and Ethics*, 3:1–23, 06 2022.

[4] Nafis Faizi and Yasir Alvi. Correlation. In *Biostatistics Manual for Health Research*, pages 109–126. ScienceDirect, 2023.

[5] Stephanie Glen. Kendall's tau (kendall rank correlation coefficient).

[6] Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. *Advances in neural information processing systems*, 2016.

[7] Richard Johnson and Dean Wichern. *Applied Multivariate Statistical Analysis*. Pearson, 1993.

[8] Hamid Karimi, Muhammad Fawad Akbar Khan, Haochen Liu, Tyler Derr, and Hui Liu. Enhancing individual fairness through propensity score matching. In *2022 IEEE 9th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 1–10, 2022.

[9] Junpei Komiyama, Akiko Takeda, Junya Honda, and Hajime Shimao. Nonconvex optimization for regression with fairness constraints. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2737–2746. PMLR, 10–15 Jul 2018.

[10] Nikita Kozodoi, Johannes Jacob, and Stefan Lessmann. Fairness in credit scoring: Assessment, implementation and profit implications. *European Journal of Operational Research*, 2022.

[11] Michael Kutner, Christopher Nachtsheim, John Neter, and William Li. *Applied Linear Statistical Models*. 1974.

[12] Lily Morse, Gerald Kane, and Yazeed Awwad. Protected attributes and "fairness through unawareness", 2020.

[13] MM Mukaka. A guide to appropriate use of correlation coefficient in medical research, 2012.

[14] James A. Nichols, Hsien W. Herbert Chan, and Matthew Baker. Machine learning: applications of artificial intelligence to imaging and diagnosis, 2018.

[15] Luca Oneto and Silvia Chiappa. Fairness in machine learning. In *Recent trends in learning from data: Tutorials from the inns big data and deep learning conference (innsbddl2019)*, 2020.

[16] Llukan Puka. Kendall's tau, 2014.

[17] Iqbal Sarker. Machine learning: Algorithms, real-world applications and research directions. 2021.

[18] Marco Scutari. fairml: A statistician's take on fair machine learning modelling. *arXiv preprint arXiv:2305.02009*, 2023.

[19] Robert Wolpert. Institute of statistics and decision sciences, 2009.

[20] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rogriguez, and Krishna P Gummadi. Fairness constraints: Mechanisms for fair classification. In *Artificial intelligence and statistics*, pages 962–970. PMLR, 2017.